IMS ASSOCIATES, INC.

July 9, 1976

ATTENTION: 8K BASIC USERS

We have discovered that using a line in a Basic Program that is longer than 51 characters causes the LIST command to malfunction, i.e. the LIST Command will only list the program up to and including that line. The list stops after printing that line containing 51 characters. This does not affect the rest of the program. The program can be run, changed, etc. We are currently working on a solution to the above problem. As a temporary fix, do not have lines longer than 51 characters in a Basic Program.

Very truly yours,
IMSAI

Marva Van Natta

Marva Van Natta
Manager
General Services

MVNJV/tt

IMS ASSOCIATES, INC.

Dear Registered IMSAI Owner,

We are releasing our initial version of 8K BASIC to you now. The final, and improved, version will be available in about five months.

We will be happy to send you the improved, final version of 8K BASIC when it becomes available at no additional charge to you. Just fill out and mail the form below to us.

Yes, I want the final 8K BASIC when it is avaialable at no additional charge to me.

NAME ____

ADDRESS ____

CITY ____________________ STATE __________ ZIP __________

TELEPHONE (   ) ______________________________________

14860 WICKS BLVD. SAN LEANDRO CALIFORNIA

IMSAI 8K BASIC

I. Loading (Paper tape version)

A. Code the loader in from the front panel starting at HEX 2000
B. Reset the system and put address to HEX 2000
C. Mount the paper tape in the reader starting anywhere in the leading rubout codes
D. Set system to run
E. Hit STOP, RESET, RUN
F. Hit the CR key and type: NEW CR
G. If your memory has WRITE PROTECT switches, turn the first 8K bytes off.

II. Resetting Basic

A. If system is hung up a reset may be required
B. Hit STOP, RESET, RUN
C. Type a carriage return
D. System types 'READY'
E. Program will not be lost. You may list, run or modify it.
F. If ROM version is used and power is frist turned on you must type the command, NEW before any other action

III. Entering a program from the keyboard

A. Type the command, NEW to delete the previous program
B. Enter the program one line at a time, type a CR after each line
C. Each statement must be numbered, but they need not be typed in order. The program will be listed or executed in line number order
D. A line may be deleted by typing its line number followed by a CR
E. A line may be replaced by typing the same line number followed by the updated statement and CR
F. A line may be inserted by typing a new, unique line number, followed by the statement and CR
G. A line may be retyped if a mistake is noticed before the CR is typed by typing control C
H. The last character may be deleted by typing the RUBOUT key. A back slash, the deleted character, and another back slash will echo. Now type the correct character and continue

IV. Entering a program from paper tape

A. Type the command NEW to delete the previous program
B. Mount the paper tape in the reader staring anywhere in the Rubout Leader
C. Type the command TAPE
D. Manually start the reader

V. Listing the program

A. Type the command LIST

VI. Punching the program in paper tape

A. Type PUNCH

B. The program will be punched and listed

VII. Running the program

A. Type the command: RUN

B. The program will run to completion or until an error is encountered, or a STOP statement is executed

C. The program may be terminated by typing: Control C

D. Input statements will prompt you with a question mark. Type the value and a CR. Multiple values may be input separated by commas.

E. Errors are of the form: XX ERR @ 1111 where XX is the error code, and 1111 is the line number. (Immediate commands have a line number = ????)

VIII. Errors and Debugging

Code	Meaning
DA	Out of data. A READ statement tried to read another DATA value and there were not more DATA statements
NX	FOR with no matching NEXT or NEXT with no matching FOR
OV	Arithmetic overflow or underflow or divide by zero. The range: $\pm$9.99999E-64 $\geq$ x $\geq$ $\pm$9.99999E+63 has been exceeded
SN	Syntax error: unidentified statement or required syntax rules have been violated
ST	Execution stack error (internal) Either: 1. BASIC was not properly initialized (NEW was not first command after loading or power up) 2. The program is too large 3. The BASIC interpreter has been destroyed and must be re-loaded
UL	Undefined line number in an IF...THEN, GOTO, or GOSUB statement
RT	Encountered RETURN and no GOSUB was in effect

A. After the error has been corrected, (if possible), the program may be restarted by typing the command: GOTO 1111 where 1111 is the line number in the error message

B. Variables may be inspected by using the PRINT (or?) command

C. Variables may be modified with the LET or INPUT commands

IX. Immediate commands

A. The following statements are commands if they are typed without a line number. They will be executed immediately and control will return to the command input routine (except for GOTO or a TRUE IF statement): PRINT (or?), INPUT, READ, LET, GOTO, & IF...THEN

B. The IF command may be typed without the THEN clause to test variables. For example: If A = B CR. If you get: UL ERR @ ???? then the relation is true otherwise it is false.

X. Statement Syntax

```
                =
                >
                <
A. If <expr>       <expr>    THEN        line
                <>
                >=
                <=
```

B. READ < list of variables separated by commas >

C. DATA < list of values separated by commas >

D. PRINT [< expr or literal enclosed in double quotes > [; , ...]]
(or ?)

E. INPUT < list of variables separated by commas >

F. [LET] < variable > = <expr>

G. GOTO < line >

H. GOSUB < line >

I. RETURN

J. FOR <variable> = <expr> TO <expr> STEP [<expr>]

K. NEXT < variable >

L. REM <comment>
(or !)

M. STOP

N. END

O. ON < expr > GOTO < list of line numbers >

P. ON < expr > GOSUB < list of line numbers >

Q. DIM < var dimension(s) >

R. DEF FN<var > (arg(s)) = <expr (arg(s))>

S. CHANGE [<array name > / <string var>] [<string var > / < array name >]

XI. Function available

RND, SQR, ABX, INT, SGN, TAB, NUM, SIN, COS, TAN, COT, ATN, EXP, LOG, ASC, CAR, INSTR, LEFT, LEN, MID, RIGHT, SPACE, STR and VAL

XII. Known Problems

1. SQR does not work for arguments less than 1

COMPATIBILITY WITH BASIC-PLUS

A. Features

1. Line numbers range from 1 to 9999 only
2. Multiple lines per statement not allowed

B. Statements

2. READ - Same
3. DATA - Same
4. PRINT - TAB() not available, literals in double quotes only
5. INPUT - Same
6. LET - Single variable assignment only
7. GOTO - Same
8. GOSUB - Same
9. RETURN - Same
10. FOR - Same
11. NEXT - Same
12. REM - Same
13. STOP - Same
14. END - not required, acts as REM
15. ON . . . GOTO - Same
16. ON . . . GOSUB - Same
17. DIM - Same
18. DEF - Same
19. CHANGE - Same

```
                    ;
                    ; *** 8K BASIC LOADER  REV 1. ***
                    ;
                    ;
                    ; TO USE THIS LOADER  FIRST KEY IT IN, STARTING
                    ; AT LOCATION 2000H.  THEN MOUNT THE TAPE IN THE READER
                    ; ON THE TELETYPE, SET THE ADDRESS SWITCHES TO 2000H,
                    ; THEN PRESS 'STOP', 'RESET', 'EXAMINE', AND 'RUN'.
                    ; THE TELETYPE READER SHOULD START AUTOMATICALLY.
                    ;
                    ; NOTE: FOR REASONS OF BREVITY, THIS LOADER DOES NOT
                    ; CHECK THE CHECKSUMS ON THE TAPE.
                    ;
0000                          ORG     2000H
                    ;
0002                TTY       EQU     2           ;TELETYPE DATA PORT
0003                TTS       EQU     3           ;TETELYPE STATUS PORT
0002                TTR       EQU     2           ;TELETYPE READY BIT
0011                RON       EQU     11H         ;READER ON BIT
                    ;
                    ; THE LOADER BEGINS HERE
                    ;
2000 3ECE           LOAD:     MVI     A,0CEH      ;GET MODE COMMAND
2002 D303                     OUT     TTS         ;ISSUE IT
2004 3E17                     MVI     A,17H       ;GET COMMAND
2006 D303                     OUT     TTS         ;ISSUE IT
2008 3E11                     MVI     A,RON       ;GET 'X-ON' CHAR
200A D302                     OUT     TTY         ;START THE READER
200C 316720                   LXI     6,PEND+10   ;SET UP THE STACK
                    ;
                    ; GO THROUGH THIS LOOP ONCE FOR EACH RECORD
                    ;
200F CD5120         LOOP1:    CALL    INCH        ;GET A CHARACTER
2012 FE3A                     CPI     ':'         ;IS IT A COLON
2014 C20F20                   JNZ     LOOP1       ;WAIT FOR COLON
2017 CD3720                   CALL    GETBT       ;GET THE COUNT
201A B7                       ORA     A           ;SET FLAGS
201B CA3620                   JZ      EOF         ;BRANCH IF EOF RECORD
201E 47                       MOV     B,A         ;ELSE, PUT COUNT INTO B REG
201F CD3720                   CALL    GETBT       ;GET HI BYTE OF ADDR
2022 67                       MOV     H,A         ; INTO H
2023 CD3720                   CALL    GETBT       ;GET LO BYTE OF ADDRESS
2026 6F                       MOV     L,A         ; INTO L
2027 CD3720                   CALL    GETBT       ;GET TYPE BYTE AND IGNORE
                    ;
                    ; GO THROUGH THIS LOOP ONCE FOR EACH DATA BYTE IN
                    ; A RECORD
                    ;
202A CD3720         LOOP2:    CALL    GETBT       ;GET A DATA BYTE
202D 77                       MOV     M,A         ;STORE IT
202E 23                       INX     H           ;BUMP ADDRESS
202F 05                       DCR     B           ;DECREMENT COUNT
2030 C22A20                   JNZ     LOOP2       ;DO IT AGAIN
2033 C30F20                   JMP     LOOP1       ;GO GET NEXT RECORD
                    ;
2036 76             EOF:      HLT
```

```
                  ;
                  ;
                  ; THIS ROUTINE READS TWO CHARACTERS FROM
                  ; THE TAPE, AND ASSEMBLES THEM INTO A BYTE, WHICH
                  ; IS RETURNED IN THE A REGISTER
                  ;
2037 CD4420       GETBT:  CALL    INDIG   ;GET A DIGITI
203A 87                   ADD     A       ;SHIFT IT ONE BIT
203B 87                   ADD     A       ;   TWO BITS
203C 87                   ADD     A       ;   THREE BITS
203D 87                   ADD     A       ; AND FOUR BITS
203E 57                   MOV     D,A     ;SAVE IT IN D
203F CD4420               CALL    INDIG   ;GET ANOTHER DIGIT
2042 B2                   ORA     D       ;OR IN LAST DIGIT
2043 C9                   RET             ;AND RETURN
                  ;
                  ; THIS ROUTINE READS A HEX DIGIT FROM THE
                  ; TAPE. NOTE THAT IT DOES NO VALIDITY CHECKING.
                  ;
2044 CD5120       INDIG:  CALL    INCH    ;GET A CHAR FROM THE TAPE
2047 FE3A                 CPI     '9'+1   ;CHECK FOR NUMERIC
2049 FA4E20               JM      IND1    ;SKIP IF NUMERIC
204C C609                 ADI     9       ;ELSE FUDGE, SO 'A'-'F' => 4A -
204E E60F         IND1:   ANI     0FH     ;MASK OFF LO 4 BITS
2050 C9                   RET             ;AND RETURN
                  ;
                  ; THIS ROUTIEN READS A CHARACTER FROM THE TELETYPE
                  ; PAPER TAPE READER
                  ;
2051 DB03         INCH:   IN      TTS     ;GET TELETYPE STATUS
2053 E602                 ANI     TTR     ; WAIT TILL READY
2055 CA5120               JZ      INCH
2058 DB02                 IN      TTY     ;GET THE CHAR
205A E67F                 ANI     7FH     ;KILL THE PARITY BIT
205C C9                   RET
                  ;
205D              PEND    EQU     $
0000                      END


2000 3E CE D3 03 3E 17 D3 03 3E 11 D3 02 31 67 20 CD
2010 51 20 FE 3A C2 0F 20 CD 37 20 B7 CA 36 20 47 CD
2020 37 20 67 CD 37 20 6F CD 37 20 CD 37 20 77 23 05
2030 C2 2A 20 C3 0F 20 76 CD 44 20 87 87 87 87 57 CD
2040 44 20 B2 C9 CD 51 20 FE 3A FA 4E 20 C6 09 E6 0F
2050 C9 DB 03 E6 02 CA 51 20 DB 02 E6 7F C9
```

```
ADDR OBJECT   LINE TAG    OPCD OPRNDS           COMMENTS

0000               BASIC: ORG  0
0000 210024               LXI  H,RAM+1024       ;POINT FIRST POSSIBLE END OF RAM
0003 3EFA                 MVI  A,250            ;GET MODE SET
0005 C34900               JMP  INIT1            ;GO CONTINUE
                   ;
                   ;
0008               RST1:  ORG  8
                   ;
                   ; SKIP CHARS POINTED BY H,L UNTIL NON-BLANK,
                   ; LEAVE IN REG A.
                   ;
0008 7E                   MOV  A,M              ;LOAD THE BYTE AT (H,L)
0009 FE20                 CPI  ' '              ;TEST IF BLANK
000B C0                   RNZ                   ;RETURN IF NOT
000C 23                   INX  H                ;POINT NEXT
000D C30800               JMP  RST1             ;LOOP
                   ;
                   ;
0010               RST2:  ORG  16
                   ;
                   ; COMPARE STRING AT (H,L) TO STRING AT (D,E)
                   ; RETURN IF EQUAL (THRU X'00' IN D,E) OR ON FIRST NOT EQUAL
                   ; ONLY THE FIRST THREE CHARS NEED BE EQUAL
                   ; IGNORE ALL SPACES
                   ;
0010 C5                   PUSH B                ;SAVE B,C
0011 0600                 MVI  B,0              ;INIT COUNT
0013 CF            COMP1: RST  1                ;SKIP SPACES
0014 1A                   LDAX D                ;GET CHAR TO MATCH WITH
0015 C34D1A               JMP  COMP2            ;CONTINUE ELSEWHERE
                   ;
                   ;
0018               RST3:  ORG  24
                   ;
                   ; PRINT: 'XX ERR @ NNNN'
                   ;
0018 210520               LXI  H,IOBUF          ;POINT BUFFER
001B 70                   MOV  M,B              ;MOVE HI CHAR
001C 23                   INX  H                ;POINT NEXT
001D C30E1C               JMP  ERROR            ;CONTINUE ELSEWHERE
                   ;
                   ;
0028               RST5:  ORG  40
                   ;
                   ; LOAD THE FLOATING POINT ACCUMULATOR WITH THE 4 BYTES AT (H,
                   ;
0028 114820               LXI  D,FACC           ;POINT FLOAT ACC
002B 0604                 MVI  B,4              ;BYTE COUNT
002D C3201C               JMP  COPYH            ;GO MOVE IT
                   ;
                   ;
```

```
ADDR OBJECT   LINE TAG    OPCD OPRNDS          COMMENTS

0030          0097 RST6:  ORG  48
              0098 ;
              0099 ; STORE THE FLOATING POINT ACCUMULATOR AT (H,L)
              0100 ;
0030 114820   0101        LXI  D,FACC          ;POINT FLOAT ACC
0033 0604     0102        MVI  B,4             ;BYTE COUNT
0035 C3241C   0103        JMP  COPYD           ;GO MOVE IT
              0104 ;
              0105 ;
0038          0106 RST7:  ORG  56
              0107 ;
              0108 ; INCREMENT H,L BY BYTE AT (SP), RETURN TO (SP)+1
              0109 ;
0038 E3       0110        XTHL                 ;GET RETURN ADDR IN H,L
0039 7E       0111        MOV  A,M             ;GET THE INCREMENT
003A 23       0112        INX  H               ;POINT TRUE RETURN
003B E3       0113        XTHL                 ;PUT BACK TO STACK
003C D5       0114        PUSH D               ;SAVE D,E
003D 5F       0115        MOV  E,A             ;PUT IN LOW
003E B7       0116        ORA  A               ;TEST SIGN
003F 1600     0117        MVI  D,0             ;DEFAULT POSITIVE
0041 F24600   0118        JP   RST7A           ;BRIF +
0044 16FF     0119        MVI  D,255           ;ELSE, NEG
0046 19       0120 RST7A: DAD  D               ;BUMP H,L
0047 D1       0121        POP  D               ;RESTORE D,E
0048 C9       0122        RET                  ;RETURN
```

```
ADDR OBJECT   LINE TAG     OPCD OPRNDS              COMMENTS

                0124 ;
                0125 ;  INITIALIZATION ROUTINE
                0126 ;  DETERMINE MEMORY SIZE. (START AT 9K AND TRY 1K INCREMENTS T
                0127 ;  SETUP POINTERS FOR STACK, DATA, AND PROGRAM
                0128 ;  INIT SIO BOARD
                0129 ;
0049 D303       0130 INIT1:  OUT  TTY+1              ;WRITE TO SIO
004B 3E17       0131         MVI  A,23               ;CMMD: DTR, ENABLE TRNS. & RCVR.
004D D303       0132         OUT  TTY+1              ;WRITE TO SIO
004F 010004     0133         LXI  B,1024             ;1K INCR
0052 7E         0134 INIT2:  MOV  A,M                ;GET A BYTE FROM MEMORY
0053 2F         0135         CMA                     ;COMPLEMENT
0054 77         0136         MOV  M,A                ;REPLACE
0055 BE         0137         CMP  M                  ;TEST IF RAM/ROM/END
0056 C25F00     0138         JNZ  INIT3              ;BRIF OUT OF RAM
0059 2F         0139         CMA                     ;RE-COMPLEMENT
005A 77         0140         MOV  M,A                ;PUT ORIG BACK
005B 09         0141         DAD  B                  ;POINT NEXT BLOCK
005C D25200     0142         JNC  INIT2              ;LOOP
005F F9         0143 INIT3:  SPHL                    ;SET STACK POINTER TO END OF MEM
0060 0100FF     0144         LXI  B,-256             ;ALLOW 256 BYTES FOR DYNAMIC WOR
0063 09         0145         DAD  B                  ;ADD TO ADDRESS
0064 22F821     0146         SHLD DATAB              ;SAVE ADDR OF START OF DATA
0067 AF         0147         XRA  A                  ;GET A ZERO IN A
0068 F5         0148         PUSH PSW                ;SET STACK 1 LEVEL DEEP WITHOUT
0069 210000     0149         LXI  H,0                ;CLEAR H,L
006C 39         0150         DAD  SP                 ;SP TO H,L
006D 228021     0151         SHLD STACK              ;SAVE BEG OF STACK
0070 328221     0152         STA  FORNE              ;CLEAR FOR/NEXT TABLE
0073 326621     0153         STA  TAPES              ;SET NOT FROM TAPE
0076 326821     0154         STA  DIMSW              ;SET NORMAL EXPR
0079 32F321     0155         STA  OUTSW              ;SET NOT OUTPUT SUPPRESS
007C 326021     0156         STA  ILSW               ;SET NOT INPUT LINE
007F 214422     0157         LXI  H,BEGPR            ;GET ADDRESS OF BEGIN OF PROG AR
0082 77         0158         MOV  M,A                ;CLEAR BYTE
0083 115C1D     0159         LXI  D,NRNDX            ;POINT RND CTRS
0086 216C21     0160         LXI  H,TRNDX            ;POINT UPDATED CTRS
0089 0608       0161         MVI  B,8                ;LOOP CTR
008B CD241C     0162         CALL COPYD              ;GO MOVE IT
008E 0608       0163         MVI  B,8                ;LOOP CTR
0090 115C1D     0164         LXI  D,NRNDX            ;POINT BACK
0093 CD241C     0165         CALL COPYD              ;GO MOVE IT
0096 3603       0166         MVI  M,3                ;INIT RND SW
```

```
ADDR OBJECT    LINE TAG    OPCD OPRNDS           COMMENTS

0098           0168 RDY:   EQU  $
               0169 ;
               0170 ; PRINT 'READY'
               0171 ; IF INIT, THEN GENERATE RANDOM NUMBERS UNTIL SOME ACTION ON
               0172 ; TTY KEYBOARD (LIKE CR)
               0173 ;
0098 21CA1E    0174        LXI  H,READY          ;POINT READY MSG
009B CD9119    0175        CALL TERMM            ;GO PRINT IT
009E 3A7C21    0176 RDY1:  LDA  RNDSW            ;GET INIT SW
00A1 FE03      0177        CPI  3                ;TEST IF 3 (INIT)
00A3 C2B600    0178        JNZ  GETCM            ;BRIF NOT
00A6 DB03      0179        IN   TTY+1            ;GET TTY STATUS
00A8 E602      0180        ANI  2                ;ISOLATE RXRDY
00AA C2B300    0181        JNZ  RDY2             ;BRIF PERSON IS READY
00AD CDE411    0182        CALL RND              ;GET A RANDOM NUMBER
00B0 C39E00    0183        JMP  RDY1             ;LOOP
00B3 327C21    0184 RDY2:  STA  RNDSW            ;SET SW=NOT INIT; AND RANDOMIZE
```

```
ADDR OBJECT   LINE TAG    OPCD OPRNDS          COMMENTS

00B6               GETCM:  EQU  $
                   ;
                   ;
                   ; COMMAND INPUT ROUTINE
                   ;
                   ; READ A LINE FROM THE TTY
                   ; IF STARTS WITH NUMERIC CHARACTERS, ASSUME IT'S A BASIC STAT
                   ; IF NOT, THEN IT IS EITHER AN IMMEDIATE STATEMENT, OR A COMM
                   ;
00B6 3E3A                  MVI  A,':'           ;PROMPT & ON SET FOR SW
00B8 326921                STA  EDSW            ;SET MODE=EDIT
00BB 2A8021                LHLD STACK           ;GET STACK ADDRESS
00BE F9                    SPHL                 ;SET REG SP
00BF CD0518                CALL TERMI           ;GET A LINE
00C2 CDC01A                CALL PACK            ;GO PACK THE NUMBER INTO B,C
00C5 78                    MOV  A,B             ;GET HI BYTE OF LINE NUMBER
00C6 B1                    ORA  C               ;PLUS LOW BYTE
00C7 CA5001                JZ   EXEC            ;BRIF EXEC STATEMENT
00CA C5                    PUSH B               ;SAVE LINE NUMBER
00CB 11FD21                LXI  D,IMMED+1       ;POINT SAVE AREA
00CE EB                    XCHG                 ;FLIP/FLOP
00CF 70                    MOV  M,B             ;PUT LO LINE
00D0 23                    INX  H               ;POINT NEXT
00D1 71                    MOV  M,C             ;PUT LO LINE
00D2 23                    INX  H               ;POINT NEXT
00D3 0603                  MVI  B,3             ;INIT COUNT
00D5 1A            EDIT1:  LDAX D               ;GET A BYTE
00D6 77                    MOV  M,A             ;PUT IT DOWN
00D7 04                    INR  B               ;COUNT IT
00D8 23                    INX  H               ;POINT NEXT
00D9 13                    INX  D               ;DITTO
00DA B7                    ORA  A               ;TEST BYTE JUST MOVED
00DB C2D500                JNZ  EDIT1           ;LOOP
00DE 78                    MOV  A,B             ;GET COUNT
00DF 32FC21                STA  IMMED           ;STORE THE COUNT
00E2 C1                    POP  B               ;GET LINE NUM
00E3 214522                LXI  H,BEGPR         ;POINT BEGINNING OF PROGRAM
00E6 7E            EDIT2:  MOV  A,M             ;GET LEN CODE
00E7 E5                    PUSH H               ;SAVE ADDR
00E8 B7                    ORA  A               ;TEST IT
00E9 CA1F01                JZ   EDIT5           ;BRIF END
00EC 23                    INX  H               ;POINT HI LINE
00ED 7E                    MOV  A,M             ;LOAD IT
00EE B8                    CMP  B               ;COMPARE
00EF DA1201                JC   EDIT4           ;BRIF LOW
00F2 C21F01                JNZ  EDIT5           ;EDIT5 BRIF HIGH
00F5 23                    INX  H               ;POINT LO LINE
00F6 7E                    MOV  A,M             ;LOAD IT
00F7 B9                    CMP  C               ;COMPARE
00F8 DA1201                JC   EDIT4           ;BRIF LOW
00FB C21F01                JNZ  EDIT5           ;BRIF HIGH
```

```
ADDR OBJECT   LINE TAG     OPCD OPRNDS          COMMENTS

00FE 2B       0237         DCX  H               ;POINT BACK
00FF 2B       0238         DCX  H               ;TO BEGIN
0100 54       0239         MOV  D,H             ;COPY ADDR
0101 5D       0240         MOV  E,L             ;TO D,E
0102 0600     0241         MVI  B,0             ;GET A ZERO
0104 4E       0242         MOV  C,M             ;GET LEN
0105 09       0243         DAD  B               ;POINT NEXT STMT
0106 7E       0244 EDIT3:  MOV  A,M             ;GET LEN NEXT STMT
0107 B7       0245         ORA  A               ;TEST IT
0108 CA1A01   0246         JZ   EDIT4           ;BRIF END
010B 47       0247         MOV  B,A             ;SET LENGTH
010C CD2D1C   0248         CALL COPYH           ;ELSE MOVE LINE
010F C30601   0249         JMP  EDIT3           ;LOOP
0112 E1       0250 EDIT4:  POP  H               ;GET ADDR
0113 1600     0251         MVI  D,0             ;ZERO HI LEN
0115 5E       0252         MOV  E,M             ;GET LO LEN
0116 19       0253         DAD  D               ;COMPUTE ADDR NEXT LINE
0117 C3E600   0254         JMP  EDIT2           ;LOOP
011A EB       0255 EDIT4:  XCHG                 ;PUT NEW ADDR TO H,L
011B 77       0256         MOV  M,A             ;MARK END
011C 22FA21   0257         SHLD PROGE           ;AND UPDATE ADDRESS
011F 3AFC21   0258 EDIT5:  LDA  IMMED           ;GET LEN OF INSERT
0122 FE04     0259         CPI  4               ;TEST IF DELETE
0124 CAB600   0260         JZ   GETCM           ;BRIF IS
0127 4F       0261         MOV  C,A             ;SET LO LEN
0128 0600     0262         MVI  B,0             ;ZERO HI LEN
012A 2AFA21   0263         LHLD PROGE           ;GET END OF PROG
012D 54       0264         MOV  D,H             ;COPY TO
012E 5D       0265         MOV  E,L             ;D,E
012F 09       0266         DAD  B               ;DISP LEN OF INSERT
0130 22FA21   0267         SHLD PROGE           ;UPDATE END POINT
0133 C1       0268         POP  B               ;GET ADDR
0134 1A       0269 EDIT6:  LDAX D               ;GET A BYTE
0135 77       0270         MOV  M,A             ;COPY IT
0136 1B       0271         DCX  D               ;POINT PRIOR
0137 2B       0272         DCX  H               ;DITTO
0138 7A       0273         MOV  A,D             ;GET HI ADDR
0139 B8       0274         CMP  B               ;COMPARE
013A CA4001   0275         JZ   EDIT7           ;BRIF HI EQUAL
013D D23401   0276         JNC  EDIT6           ;BRIF NOT LESS
0140 7B       0277 EDIT7:  MOV  A,E             ;GET LO ADDR
0141 B9       0278         CMP  C               ;COMPARE
0142 D23401   0279         JNC  EDIT6           ;BRIF NOT LESS
0145 13       0280         INX  D               ;POINT FORWARD
0146 21FC21   0281         LXI  H,IMMED         ;POINT INSERT
0149 46       0282         MOV  B,M             ;GET LENGTH
014A CD2D1C   0283         CALL COPYH           ;GO MOVE IT
014D C3B600   0284         JMP  GETCM           ;GO GET ANOTHER COMMAND
```

```
ADDR OBJECT   LINE TAG     OPCD OPRNDS              COMMENTS

0150               EXEC:   EQU  $
                   ;
                   ;
                   ;
                   ;  DECODE COMMAND IN IOBUFF
                   ;  EXECUTE IF POSSIBLE
                   ;  THEN GOTO GET NEXT COMMAND
                   ;
                   ;
0150 326521                STA  MULTI               ;RESET MULTI SW
0153 327021                STA  FNMOD               ;RESET FN TYPE
0156 3C                    INR  A                   ;GET A ONE
0157 326721                STA  RUNSW               ;SET IMMEDIATE MODE
015A 210520                LXI  H,IOBUF             ;POINT SMT
015D 11FC21                LXI  D,IMMED             ;POINT NEW AREA
0160 7E            EXEC1:  MOV  A,M                 ;GET A BYTE
0161 12                    STAX D                   ;PUT TO (D,L)
0162 13                    INX  D                   ;POINT NEXT
0163 23                    INX  H                   ;DITTO
0164 B7                    ORA  A                   ;TEST BYTE
0165 C26001                JNZ  EXEC1               ;CONTINUE
0168 21401E                LXI  H,NULLI             ;POINT NO LINE NUM
016B 227E21                SHLD LINE                ;SAVE ADDR
016E 21FC21                LXI  H,IMMED             ;POINT START OF CMMD
0171 C32002                JMP  RUN3                ;GO INTO RUN PROCESSOR
```



```
ADDR OBJECT   LINE TAG     OPCD OPRNDS              COMMENTS

0174               SAVE:   EQU  $
                   ;
                   ;  SAVE COMMAND, TURN THE PUNCH ON
                   ;  THEN LIST THE PROGRAM
                   ;  LAST LINE IS 'KEY' COMMAND TO RESET ON INPUT
                   ;
0174 3E02                  MVI  A,2                 ;SET SW
0176 326621                STA  TAPES               ;= PUNCH
0179 3E12                  MVI  A,13                ;GET DC2 (PUNCH ON)
017B CD2719                CALL TESTO               ;GO WRITE IT
017E CDB41A                CALL HDRTL               ;GO PUT HEADER
0181 C35102                JMP  LIST                ;GO PROCESS AS IF LIST
```



```
ADDR OBJECT   LINE TAG     OPCD OPRNDS              COMMENTS

0184               TAPE:   EQU  $
                   ;
                   ;  TAPE COMMAND
                   ;  SET SWITCH = NO INPUT ECHO, THEN TURN ON PAPER TAPE READER
                   ;  CONTINUE UNTIL 'KEY' COMMAND
                   ;
0184 3E01                  MVI  A,1                 ;SET SW
0186 326621                STA  TAPES               ;TURN IT ON
0189 3E11                  MVI  A,17                ;GET DC1 (READER ON)
018B CD2719                CALL TESTO               ;GO WRITE IT
018E C3B600                JMP  GETCM               ;CONTINUE
```

```
IMSAI 8080 8K BASIC VER=1.3                        10:09:22  15 APR 1976  PAGE 11

ADDR OBJECT   LINE TAG    OPCD OPRNDS              COMMENTS

0191               KEY:   EQU  $
                   ;
                   ; KEY COMMAND
                   ; RESET SWITCH, TURN READER OFF
                   ;
0191 AF                   XRA  A                   ;CLEAR A
0192 326621               STA  TAPES               ;RESET SW
0195 212E1E               LXI  H,PCHOF             ;POINT DC3/DC4 (RDR OFF/PCH OFF)
0198 CD9119               CALL TERMM               ;GO WRITE TO TTY
019B C39800               JMP  RDY                 ;CONTINUE
```

```
IMSAI 8080 8K BASIC VER=1.3                        10:09:27  15 APR 1976  PAGE 12

ADDR OBJECT   LINE TAG    OPCD OPRNDS              COMMENTS

019E               NEW:   EQU  $
                   ;
                   ; NEW COMMAND
                   ; DELETE OLD PROGRAM
                   ; CLEAR OLD DATA VALUES
                   ;
019E 214522               LXI  H,BEGPR             ;POINT BEGINNING OF PROG
01A1 AF                   XRA  A                   ;GET A ZERO
01A2 77                   MOV  M,A                 ;MARK PROGRAM NULL
01A3 22FA21               SHLD PROGE               ;MARK END
01A6 2AF821               LHLD DATAB               ;POINT BEGINNING OF DATA
01A9 77                   MOV  M,A                 ;DELETE ALL VARIABLES
01AA C3B600               JMP  GETCM               ;GO NEXT COMMAND
```

```
IMSAI 8080 8K BASIC VER=1.3                        10:09:35  15 APR 1976  PAGE 13

ADDR OBJECT   LINE TAG    OPCD OPRNDS              COMMENTS

01AD               FREE:  EQU  $
                   ;
                   ; FREE COMMAND
                   ; COMPUTE AMOUNT OF AVAILABLE STORAGE (EXCLUDING DATA AREA)
                   ;
01AD 2AF821               LHLD DATAB               ;GET DATA BEG ADDRESS
01B0 EB                   XCHG                     ;PUT IN D,E
01B1 2AFA21               LHLD PROGE               ;GET PROGRAM END ADDRESS
01B4 7B                   MOV  A,E                 ;LO ADDR TO REG A
01B5 95                   SUB  L                   ;SUBTRACT
01B6 5F                   MOV  E,A                 ;SAVE IT
01B7 7A                   MOV  A,D                 ;HI ADDR TO REG A
01B8 9C                   SBB  H                   ;SUBTRACT
01B9 57                   MOV  D,A                 ;SAVE IT
01BA CD471D               CALL BINFL               ;GO FLOAT D,E
01BD 210520               LXI  H,IOBUF             ;POINT BUFFER
01C0 CDF308               CALL FOUT                ;GO CONVERT TO OUTPUT
01C3 3600                 MVI  M,0                 ;MARK END
01C5 CD3919               CALL TERMO               ;GO WRITE IT
01C8 C3B600               JMP  GETCM               ;CONTINUE
```

```
ADDR OBJECT   TAG     OPCD OPRNDS        COMMENTS

01CB          RUNCM:  EQU  $
              ;
              ; RUN PROCESSOR. GET NEXT STATMENT. AND EXECUTE IT
              ; IF IN IMMEDIATE MODE. THEN RETURN TO GETCMMD
              ;
              ;
01CB AF               XRA  A             ;CLEAR A REG
01CC 326921           STA  EDSW          ;TURN OFF EDIT MODE
01CF 326721           STA  RUNSW         ;RESET SWITCH
01D2 328221           STA  FORNE         ;INIT THE FOR/NEXT TABLE
01D5 327C21           STA  RNDSW         ;INIT THE RND SEED
01D8 2AF821           LHLD DATAB         ;GET ADDR OF DATA POOL
01DB 77               MOV  M,A           ;CLEAR IT
01DC 115C1D           LXI  D,NRNDX       ;POINT INIT RND PARAMS
01DF 217421           LXI  H,RNDX        ;POINT RECEIVE
01E2 0608             MVI  B,8           ;LOOP CTR
01E4 CD241C           CALL COPYD         ;MOVE THEM
01E7 214422           LXI  H,BEGPR       ;POINT 1 PRIOR TO BEGIN.
01EA 22F621           SHLD DATAP         ;RESTORE DATA STMT POINTER
01ED 23               INX  H             ;POINT TO START
01EE 226121           SHLD STMT          ;SAVE IT
01F1 C30E02           JMP  RUN2          ;GO PROCESS IT
01F4 216521   RUN:    LXI  H,MULTI       ;POINT MULTIPLE SWITCH
01F7 7E               MOV  A,M           ;GET SW
01F8 B7               ORA  A             ;TEST IT
01F9 CA0402           JZ   RUN1          ;BRIF NOT ON
01FC 3600             MVI  M,0           ;ELSE, RESET IT
01FE 2A6321           LHLD ENDLI         ;GET ADDRESS
0201 C32002           JMP  RUN3          ;GO PROCESS REMAIN
0204 2A6121   RUN1:   LHLD STMT          ;ELSE, GET ADDR OF PREV STMT
0207 5E               MOV  E,M           ;GET LEN CODE
0208 1600             MVI  D,0           ;CLEAR HIGH BYTE OF ADDR
020A 19               DAD  D             ;INCR STMT POINTER
020B 226121           SHLD STMT          ;SAVE IT
020E 3A6721   RUN2:   LDA  RUNSW         ;GET RUN TYPE
0211 B7               ORA  A             ;TEST IT
0212 C2B600           JNZ  GETCM         ;BRIF IMMEDIATE MODE
0215 7E               MOV  A,M           ;GET LEN CODE
0216 B7               ORA  A             ;TEST IF END
0217 CAF208           JZ   END           ;BRIF IS
021A 23               INX  H             ;POINT LINE NUMBER
021B 227E21           SHLD LINE          ;SAVE ADDR
021E 23               INX  H             ;POINT 2ND BYTE
021F 23               INX  H             ;POINT 1ST PGM BYTE
0220 CF       RUN3:   RST  1             ;SKIP BLANKS
0221 221220   RUN4:   SHLD ADDR1         ;SAVE ADDR
0224 CD0E1A           CALL TSTCC         ;GO SEE IF CONTROL-C OR O
0227 11F01E           LXI  D,JMPTB       ;POINT TO TABLE
022A 1A       RUN5:   LDAX D             ;GET FIRST BYTE OF LIT
022B B7               ORA  A             ;TEST IF END OF TABLE
022C CA4B02           JZ   RUN7          ;BRIF IS
```



```
ADDR OBJECT   TAG     OPCD OPRNDS        COMMENTS

022F 2A1220           LHLD ADDR1         ;GET ADDRESS OF CMMD
0232 D7               RST  2             ;GO COMPARE
0233 C24202           JNZ  RUN6          ;BRIF NOT EQUAL
0236 E5               PUSH H             ;SAVE H,L
0237 CD041B           CALL SKP2Z         ;FIND END OF LIT
023A 13               INX  D             ;POINT NEXT BYTE
023B 1A               LDAX D             ;LOAD IT
023C 6F               MOV  L,A           ;LOW BYTE TO L
023D 13               INX  D             ;POINT NEXT BYTE
023E 1A               LDAX D             ;LOAD IT
023F 67               MOV  H,A           ;HIGH BYTE TO H
0240 E3               XTHL               ;H,L TO STACK, STACK TO H,L
0241 C9               RET                ;JUMP TO PROPER ROUTINE
0242 CD041B   RUN6:   CALL SKP2Z         ;FIND END OF LIT
0245 13               INX  D             ;POINT NEXT
0246 13               INX  D             ;DITTO
0247 13               INX  D             ;POINT FIRST BYTE NEXT LIT
0248 C32A02           JMP  RUN5          ;LOOP
024B 2A1220   RUN7:   LHLD ADDR1         ;RESTORE H,L POINTER
024E C30E06           JMP  LET           ;ASSUME IT'S LET STMT
```

```
ADDR OBJECT    TAG     OPCD OPRNDS        COMMENTS

0251           LIST:   EQU  $
               ;
               ;
               ; LIST PROCESSOR
               ; DUMP THE SOURCE PROGRAM TO TTY OR PAPER TAPE
               ;
               ;
0251 CF                RST  1             ;SKIP TO NON BLANK
0252 110000            LXI  D,0           ;GET A ZERO IN D
0255 EB                XCHG               ;FLIP TO H,L
0256 224020            SHLD LINEL         ;SAVE IT
0259 219999            LXI  H,39321       ;GET HIGH NUMBER IN H,L
025C 224220            SHLD LINEH         ;SAVE IT
025F EB                XCHG               ;FLIP BACK
0260 B7                ORA  A             ;TEST IF EOL
0261 CA8102            JZ   LIST1         ;BRIF IT IS
0264 CDC01A            CALL PACK          ;GO PACK THE NUMBER, IF ANY
0267 50                MOV  D,B           ;COPY NUMBER TO D,L
0268 59                MOV  E,C           ;SAME
0269 EB                XCHG               ;FLIP TO H,L
026A 224020            SHLD LINEL         ;SAVE IT
026D 224220            SHLD LINEH         ;SAME
0270 EB                XCHG               ;RESTORE H,L
0271 CF                RST  1             ;SKIP TO NON BLANK
0272 FE2C              CPI  ','           ;TEST IF COMMA
0274 C28102            JNZ  LIST1         ;BRIF NOT
0277 23                INX  H             ;POINT NEXT
0278 CF                RST  1             ;SKIP TO NON-BLANK
0279 CDC01A            CALL PACK          ;ELSE, GO GET THE NUMBER
027C 60                MOV  H,B           ;COPY TO
027D 69                MOV  L,C           ;D,L
027E 224220            SHLD LINEH         ;SAVE IT
0281 214522    LIST1:  LXI  H,BEGPR       ;POINT BEGINNING OF PROGRAM
0284 CD0E1A    LIST2:  CALL TSTCC         ;GO SEE IF CONTROL-C OR CONTROL-
0287 7E                MOV  A,M           ;GET LEN CODE
0288 B7                ORA  A             ;TEST IF END OF PROGRAM
0289 CAF208            JZ   END           ;BRIF END OF PGM
028C D603      LIST3:  SUI  3             ;SUBTRACT THREE
028E 47                MOV  B,A           ;SAVE LEN
028F 23                INX  H             ;POINT HIGH BYTE OF LINE#
0290 EB                XCHG               ;FLIP H,L TO D,E
0291 2A4020            LHLD LINEL         ;GET LOW LINE TO TEST
0294 EB                XCHG               ;RESTORE H,L
0295 7E                MOV  A,M           ;GET LOW BYTE OF LINE NUMBER
0296 BA                CMP  D             ;COMP WITH LINEL
0297 DAD402            JC   LIST8         ;BRIF LESS
029A C2A402            JNZ  LIST4         ;BRIF NOT EQUAL
029D 23                INX  H             ;POINT NEXT
029E 7E                MOV  A,M           ;GET NEXT BYTE OF LINE#
029F 2B                DCX  H             ;POINT BACK
02A0 BB                CMP  E             ;COMP LOW BYTES
```



```
ADDR OBJECT    TAG     OPCD OPRNDS        COMMENTS

02A1 DAD402            JC   LIST8         ;BRIF LESS
02A4 EB        LIST4:  XCHG               ;SAVE H,L IN D,E
02A5 2A4220            LHLD LINEH         ;GET HIGH LINE FOR TEST
02A8 EB                XCHG               ;RESTORE H,L
02A9 7E                MOV  A,M           ;GET LINE BYTE
02AA BA                CMP  D             ;COMPARE HIGH BYTES
02AB CAB402            JZ   LIST5         ;BRIF EQUAL
02AE D2F208            JNC  END           ;BRIF HIGHER
02B1 C3BE02            JMP  LIST6         ;GO AROUND
02B4 23        LIST5:  INX  H             ;POINT NEXT
02B5 7E                MOV  A,M           ;GET NEXT BYTE
02B6 2B                DCX  H             ;POINT BACK
02B7 BB                CMP  E             ;COMPARE LOW BYTES
02B8 CABE02            JZ   LIST6         ;BRIF EQUAL
02BB D2F208            JNC  END           ;BRIF HIGHER
02BE 110520    LIST6:  LXI  D,IOBUF       ;POINT BUFFER AREA
02C1 CDDD19            CALL LINEO         ;CONVERT LINE NUMBER
02C4 7E        LIST7:  MOV  A,M           ;GET A BYTE
02C5 12                STAX D             ;PUT IT TO BUFFER
02C6 13                INX  D             ;POINT NEXT BUFF
02C7 23                INX  H             ;POINT NEXT PROG
02C8 05                DCR  B             ;DECR CTR
02C9 C2C402            JNZ  LIST7         ;LOOP
02CC E5                PUSH H             ;SAVE HL ADDR
02CD CD8919            CALL TERMO         ;GO TYPE IT
02D0 E1                POP  H             ;RETREIVE H ADDR
02D1 C38402            JMP  LIST2         ;CONTINUE
02D4 58        LIST8:  MOV  E,B           ;PUT LEN  IN E
02D5 1600              MVI  D,0           ;CLEAR D
02D7 19                DAD  D             ;POINT NEXT STMT
02D8 23                INX  H             ;POINT NEXT
02D9 23                INX  H             ;POINT LEN CODE
02DA C38402            JMP  LIST2         ;GO LIST IT
```

```
IMSAI 8080 8K BASIC VER=1.3                          10:11:59  15 APR 1976  PAGE 18

ADDR OBJECT    LINE TAG    OPCD OPRNDS               COMMENTS

02DD           0540 CONTI:  EQU  $
               0541 ;
               0542 ;  CONT AT STATEMENT FOLLOWING STOP OR AT STATEMENT WHEN CONTR
               0543 ;
               0544 ;
02DD 210020    0545         LXI  H,LINEN             ;POINT LINE NUMBER OF LAST STOP/
02E0 7E        0546         MOV  A,M                 ;GET 1ST CHAR
02E1 B7        0547         ORA  A                   ;TEST IF IMMED CMMD
02E2 C2E802    0548         JNZ  GOTO                ;BRIF NOT
02E5 C30E06    0549         JMP  LET                 ;BRIF WAS
```

```
IMSAI 8080 8K BASIC VER=1.3                          10:12:06  15 APR 1976  PAGE 19

ADDR OBJECT    LINE TAG    OPCD OPRNDS               COMMENTS

02E8           0551 GOTO:   EQU  $
               0552 ;
               0553 ;
               0554 ;  STMT:   GOTO NNNN
               0555 ;
02E8 AF        0556         XRA  A                   ;CLEAR REG A
02E9 326921    0557         STA  EDSW                ;RESET IMMED MODE (IF IT WAS SET
02EC 326721    0558         STA  RUNSW               ;AND RUN TYPE
               0559 ;
               0560 ;
02EF CD8C1A    0561 GOTO1:  CALL NOTEO               ;ERROR IF END-OF-LINE
02F2 CDC01A    0562         CALL PACK                ;GO GET LINE NUMBER IN B,C
02F5 CD681A    0563         CALL EOL                 ;ERROR IF NOT END-OF-LINE
02F8 214522    0564         LXI  H,BEGPR             ;POINT BEGINNING OF PROGRAM
02FB 7E        0565 GOTO2:  MOV  A,M                 ;GET LEN
02FC B7        0566         ORA  A                   ;TEST IF END OF PROGRAM
02FD CA941A    0567         JZ   ULERR               ;BRIF UNDEFINED STATEMENT
0300 23        0568         INX  H                   ;POINT NEXT
0301 7E        0569         MOV  A,M                 ;GET THE HIGH LINE NUMBER
0302 B8        0570         CMP  B                   ;TEST WITH DESIRED
0303 DA1B03    0571         JC   GOTO3               ;BRIF LOW
0306 23        0572         INX  H                   ;POINT NEXT BYTE
0307 7E        0573         MOV  A,M                 ;GET LOW LINE NUMBER
0308 2B        0574         DCX  H                   ;POINT BACK
0309 B9        0575         CMP  C                   ;TEST WITH WANTED
030A DA1B03    0576         JC   GOTO3               ;BRIF LOW
030D C2941A    0577         JNZ  ULERR               ;BRIF LINE MISSING
0310 2B        0578         DCX  H                   ;POINT TO START OF STMT
0311 226121    0579         SHLD STMT                ;SAVE ADDR
0314 AF        0580         XRA  A                   ;GET A ZERO
0315 326521    0581         STA  MULTI               ;TURN OFF MULTIPLE STMTS
0318 C30E02    0582         JMP  RUN2                ;GO PROCESS THE STATEMENT
031B 2B        0583 GOTO3:  DCX  H                   ;POINT START OF STMT
031C 5E        0584         MOV  E,M                 ;GET LENGTH
031D 1600      0585         MVI  D,0                 ;ZERO HI BYTE
031F 19        0586         DAD  D                   ;COMPUTE NEXT ADDR
0320 C3FB02    0587         JMP  GOTO2               ;LOOP
```

```
IMSAI 8080 8K BASIC VER=1.3                       10:13:00  15 APR 1976  PAGE 20

ADDR OBJECT   LINE TAG     OPCD OPRNDS            COMMENTS

0323          0589 RESTO:  EQU  $
              0590 ;
              0591 ;
              0592 ; STMT: RESTORE
              0593 ;
0323 CD681A   0594         CALL EOL               ;ERROR IF NOT END-OF-LINE
0326 214422   0595         LXI  H,BEGPR           ;POINT 1 BEFORE START OF PROGRAM
0329 22F621   0596         SHLD DATAP             ;FORCE NEXT DATA TO BE AT START
032C C3F401   0597         JMP  RUN               ;GO NEXT STMT
```

```
IMSAI 8080 8K BASIC VER=1.3                       10:13:05  15 APR 1976  PAGE 21

ADDR OBJECT   LINE TAG     OPCD OPRNDS            COMMENTS

032F          0599 RETUR:  EQU  $
              0600 ;
              0601 ;
              0602 ; STMT:   RETURN
              0603 ;
032F CD681A   0604         CALL EOL               ;ERROR IF NOT END-OF-LINE
0332 F1       0605         POP  PSW               ;POP THE STACK
0333 FEFF     0606         CPI  255               ;TEST IF GOSUB IN EFFECT
0335 C2A41A   0607         JNZ  RTERR             ;BRIF ERROR
0338 E1       0608         POP  H                 ;GET RETURNED STATMENT ADDRESS
0339 226121   0609         SHLD STMT              ;RESTORE
033C E1       0610         POP  H                 ;GET ENDLINE VALUE
033D 226321   0611         SHLD ENDLI             ;RESTORE
0340 F1       0612         POP  PSW               ;GET MULTI SW VALUE
0341 326521   0613         STA  MULTI             ;RESTORE
0344 C3F401   0614         JMP  RUN               ;CONTINUE (AT STMT FOLLOWING GOS
```

```
ADDR OBJECT   LINE TAG     OPCD OPRNDS           COMMENTS

0347          0616 GOSUB:  EQU  $
              0617 ;
              0618 ;
              0619 ;  STMT:   GOSUB NNNN
              0620 ;
0347 CD8C1A   0621         CALL NOTEO           ;ERROR IF END-OF-LINE
034A CDC01A   0622         CALL PACK            ;GET LINE NUMBER
034D CD681A   0623         CALL EOL             ;ERROR IF NOT END-OF-LINE
0350 3A6521   0624 GOSU1:  LDA  MULTI           ;GET SW SETTING
0353 F5       0625         PUSH PSW             ;SAVE ON STACK
0354 2A6321   0626         LHLD ENDLI           ;GET ADDR OF END OF STMT
0357 E5       0627         PUSH H               ;SAVE ONE STACK
0358 2A6121   0628         LHLD STMT            ;GET STATEMENT ADDRESS
035B E5       0629         PUSH H               ;SAVE RETURN ADDRESS IN STACK
035C 3EFF     0630         MVI  A,255           ;MARK AS GOSUB
035E F5       0631         PUSH PSW             ;SAVE STATUS
035F 214522   0632         LXI  H,BEGPR         ;POINT START OF PROGRAM
0362 C3FB02   0633         JMP  GOTO2           ;GO LOOKUP LINE AND BRANCH
```



```
ADDR OBJECT   LINE TAG     OPCD OPRNDS           COMMENTS

0365          0635 PRINT:  EQU  $
              0636 ;
              0637 ;
              0638 ;  STMT:  PRINT ....
              0639 ;
              0640 ;
0365 AF       0641         XRA  A               ;CLEAR REG A
0366 32F421   0642         STA  PRSW            ;SET SWITCH
0369 110520   0643 PRIN1:  LXI  D,IOBUF         ;POINT BUFFER
036C CF       0644         RST  1               ;SKIP TO NEXT FIELD
036D C38C03   0645         JMP  PRIN4           ;GO AROUND
0370 3EFE     0646 PRIN2:  MVI  A,254           ;SET CODE = NO CR/LF
0372 12       0647         STAX D               ;PUT TO BUFFER
0373 E5       0648         PUSH H               ;SAVE H,L
0374 CD8919   0649         CALL TERMO           ;GO PRINT IT
0377 E1       0650         POP  H               ;RESTORE H,L
0378 C36503   0651         JMP  PRINT           ;RECURSIVE TO NEXT FIELD
037B 3AF421   0652 PRIN3:  LDA  PRSW            ;GET SWITCH
037E B7       0653         ORA  A               ;TEST IF STMT ENDED WITH , OR ;
037F C2F401   0654         JNZ  RUN             ;BRIF IT DID
0382 3AF321   0655         LDA  OUTSW           ;GET CONTROL-O SW
0385 B7       0656         ORA  A               ;TEST IT
0386 CC3219   0657         CZ   CRLF            ;CR/LF IF NOT SUPPRESSED
0389 C3F401   0658         JMP  RUN             ;CONTINUE NEXT STMT
038C CD811A   0659 PRIN4:  CALL TSTEL           ;TEST IF END OF STMT
038F CAFC03   0660         JZ   PRINC           ;BRIF IT IS
0392 FE2C     0661         CPI  ','             ;TEST IF COMMA
0394 CAD303   0662         JZ   PRIN8           ;BRIF IT IS
0397 FE3B     0663         CPI  ';'             ;TEST IF SEMI-COLON
0399 CAD603   0664         JZ   PRIN9           ;BRIF IT IS
039C D5       0665         PUSH D               ;SAVE D,E
039D E5       0666         PUSH H               ;SAVE H,L
039E 116E1D   0667         LXI  D,TABLI         ;POINT LITERAL
03A1 D7       0668         RST  2               ;GO SEE IF TAB(XX)
03A2 CADF03   0669         JZ   PRINA           ;BRIF IS
03A5 E1       0670         POP  H               ;ELSE, RESTORE H,L
03A6 CD0714   0671         CALL EXPR            ;GO EVALUATE EXPRESSION
03A9 D1       0672         POP  D               ;RESTORE D,E
03AA E5       0673         PUSH H               ;SAVE H,L
03AB EB       0674         XCHG                 ;FLIP/FLOP
03AC 3AF521   0675         LDA  NS              ;GET TYPE OF RESULT
03AF FEE7     0676         CPI  231             ;TEST IF STRING
03B1 CABD03   0677         JZ   PRIN5           ;BRIF IS
03B4 CDF30B   0678         CALL FOUT            ;GO CONVERT OUTPUT
03B7 23       0679         INX  H               ;POINT NEXT
03B8 EB       0680         XCHG                 ;FLIP/FLOP
03B9 E1       0681         POP  H               ;RESTORE H,L
03BA C37003   0682         JMP  PRIN2           ;CONTINUE
03BD 115820   0683 PRIN5:  LXI  D,STRIN         ;POINT STRING
03C0 1A       0684         LDAX D               ;GET LEN
03C1 B7       0685         ORA  A               ;TEST IT
```

```
ADDR OBJECT   LINE TAG     OPCD OPRNDS          COMMENTS

03C2 CACE03   0686         JZ   PRIN7           ;BRIF NULL
03C5 47       0687         MOV  B,A             ;SAVE LEN
03C6 13       0688 PRIN6:  INX  D               ;POINT NEXT
03C7 1A       0689         LDAX D               ;GET A BYTE
03C8 77       0690         MOV  M,A             ;STORE IT
03C9 23       0691         INX  H               ;POINT NEXT
03CA 05       0692         DCR  B               ;DECR CTR
03CB C2C603   0693         JNZ  PRIN6           ;LOOP
03CE EB       0694 PRIN7:  XCHG                 ;PUT END ADDR IN D,E
03CF E1       0695         POP  H               ;RESTORE H,L
03D0 C37003   0696         JMP  PRINZ           ;CONTINUE
03D3 CDB319   0697 PRIN8:  CALL TABST           ;GO POSITION NEXT TAB
03D6 23       0698 PRIN9:  INX  H               ;POINT NEXT
03D7 3E01     0699         MVI  A,1             ;GET SETTING FOR SW
03D9 32F421   0700         STA  PRSW            ;SET THE SWITCH
03DC C36903   0701         JMP  PRIN1           ;GO NEXT FIELD
03DF D1       0702 PRINA:  POP  D               ;GET RID OF STACK ENTRY
03E0 CD0714   0703         CALL EXPR            ;GO EVALUATE
03E3 E5       0704         PUSH H               ;SAVE H,L
03E4 CD3B1C   0705         CALL FBIN            ;CONVERT TO BINARY
03E7 216B21   0706         LXI  H,COLUM         ;POINT CURRENT POSITION
03EA 96       0707         SUB  M               ;SUBTRACT (LEAVES NUMBER OF FILL
03EB E1       0708         POP  H               ;GET H,L
03EC D1       0709         POP  D               ;GET D,E
03ED FA7003   0710         JM   PRINZ           ;BRIF ALREADY PAST IT
03F0 47       0711         MOV  B,A             ;SAVE COUNT
03F1 3E20     0712         MVI  A,' '           ;GET FILL
03F3 CA7003   0713 PRINB:  JZ   PRINZ           ;BRIF COUNT ZERO
03F6 12       0714         STAX D               ;PUT ONE SPACE
03F7 13       0715         INX  D               ;POINT NEXT
03F8 05       0716         DCR  B               ;DECR CTR
03F9 C3F303   0717         JMP  PRINB           ;LOOP
03FC CD681A   0718 PRINC:  CALL EOL             ;SAVE EOL POSIT
03FF C37B03   0719         JMP  PRIN3           ;TERMINATE
```



```
ADDR OBJECT   LINE TAG     OPCD OPRNDS          COMMENTS

0402 .        0721 FOR:    EQU  $
              0722 ;
              0723 ;
              0724 ;  STMT:  FOR VAR = EXPR TO EXPR [STEP EXPR]
              0725 ;
              0726 ;
0402 CDD41B   0727         CALL VAR             ;NEXT WORD MUST BE VARIABLE
0405 EB       0728         XCHG                 ;FLIP/FLOP
0406 221820   0729         SHLD INDX            ;SAVE VARIABLE NAME
0409 EB       0730         XCHG                 ;FLIP/FLOP AGAIN
040A FE3D     0731         CPI  '='             ;TEST FOR EQUAL SIGN
040C C2A01A   0732         JNZ  SNERR           ;BRIF NO EQUAL
040F 23       0733         INX  H               ;POINT NEXT
0410 CD0714   0734         CALL EXPR            ;GO EVALUATE EXPR, IF ANY
0413 EB       0735         XCHG                 ;FLIP/FLOP AGAIN
0414 2A1820   0736         LHLD INDX            ;GET INDEX NAME
0417 EB       0737         XCHG                 ;FLIP/FLOP
0418 E5       0738         PUSH H               ;SAVE H,L
0419 CD3F1B   0739         CALL SEARC           ;GO LOCATE NAME
041C EB       0740         XCHG                 ;PUT ADDR IN H,L
041D 221220   0741         SHLD ADDR1           ;SAVE ADDR
0420 F7       0742         RST  6               ;GO STORE THE VALUE
0421 E1       0743         POP  H               ;RESTORE POINTER TO STMT
0422 116F1F   0744         LXI  D,TOLIT         ;GET LIT ADDR
0425 D7       0745         RST  2               ;GO COMPARE
0426 C2A01A   0746         JNZ  SNERR           ;BRIF ERROR
0429 CD0714   0747         CALL EXPR            ;GO EVALUATE TO-EXPR
042C E5       0748         PUSH H               ;SAVE H,L
042D 211C20   0749         LXI  H,TVAR1         ;POINT 'TO' VALUE
0430 F7       0750         RST  6               ;SAVE IT
0431 213E1E   0751         LXI  H,ONE           ;POINT CONSTANT: 1
0434 EF       0752         RST  5               ;LOAD IT
0435 E1       0753         POP  H               ;GET H,L
0436 7E       0754         MOV  A,M             ;GET THE CHAR
0437 B7       0755         ORA  A               ;TEST FOR END OF STATMENT
0438 CA4B04   0756         JZ   FOR2            ;BRIF NO STEP
043B E5       0757         PUSH H               ;RE-SAVE
043C 11311E   0758         LXI  D,STEPL         ;TEST FOR LIT 'STEP'
043F D7       0759         RST  2               ;GO COMPARE
0440 CA4704   0760         JZ   FOR1            ;BRIF STEP
0443 E1       0761         POP  H               ;RESTORE H,L
0444 C34B04   0762         JMP  FOR2            ;GO NO STEP VALUE
0447 D1       0763 FOR1:   POP  D               ;POP OFF THE STACK
0448 CD0714   0764         CALL EXPR            ;GO EVALUATE EXPRESSION
044B E5       0765 FOR2:   PUSH H               ;SAVE H,L
044C 212020   0766         LXI  H,TVAR2         ;POINT STEP VALUE
044F F7       0767         RST  6               ;SAVE IT
0450 E1       0768         POP  H               ;RESTORE H,L
0451 CD681A   0769         CALL EOL             ;ERROR IF NOT END-OF-LINE
0454 CD750F   0770         CALL FTEST           ;GET STATUS OF FACC
0457 F5       0771         PUSH PSW             ;SAVE A,STATUS
```

```
ADDR OBJECT  TAG    OPCD OPRNDS     COMMENTS

0458 211C20         LXI  H,TVAR1    ;GET END VALUE
045B EF             RST  5          ;LOAD IT
045C F1             POP  PSW        ;RESTORE STATUS
045D F26F04         JP   FOR4       ;BRIF FOR IS POSITIVE
0460 2A1220  FOR3:  LHLD ADDR1      ;GET ADDRESS OF INDEX
0463 CD530E         CALL FSUB       ;COMPARE THIS AGAINST END VALUE
0466 CA7B04         JZ   FOR5       ;BRIF START = END
0469 FA7B04         JM   FOR5       ;BRIF START > END
046C C3CF04         JMP  FOR9       ;GO LOCATE MATCHING NEXT
046F 2A1220  FOR4:  LHLD ADDR1      ;GET ADDRESS OF INDEX
0472 CD530E         CALL FSUB       ;COMPARE
0475 CA7B04         JZ   FOR5       ;BRIF START = END
0478 FACF04         JM   FOR9       ;BRIF START > END
047B 118221  FOR5:  LXI  D,FORNE    ;POINT TABLE
047E 2A1820         LHLD INDX       ;GET INDEX NAME
0481 EB             XCHG            ;FLIP/FLOP
0482 7E             MOV  A,M        ;GET COUNT
0483 47             MOV  B,A        ;STORE IT
0484 0E01           MVI  C,1        ;NEW CTR
0486 B7             ORA  A          ;TEST IF ZERO
0487 23             INX  H          ;POINT
0488 CA9E04         JZ   FOR8       ;BRIF TABLE EMPTY
048B 7E      FOR6:  MOV  A,M        ;GET 1ST BYTE
048C BA             CMP  D          ;TEST IF EQUAL
048D C29704         JNZ  FOR7       ;BRIF NOT
0490 23             INX  H          ;POINT NEXT
0491 7E             MOV  A,M        ;GET NEXT BYTE
0492 2B             DCX  H          ;POINT BACK
0493 BB             CMP  E          ;TEST IF EQUAL
0494 C29E04         JNZ  FOR8       ;BRIF EQUAL
0497 FF      FOR7:  RST  7          ;ADJUST H,L
0498 0E             DB   14
0499 0C             INR  C          ;COUNT IT
049A 05             DCR  B          ;DECR CTR
049B C28B04         JNZ  FOR6       ;LOOP
049E 79      FOR8:  MOV  A,C        ;GET UPDATED COUNT
049F FE09           CPI  9          ;TEST IF TBL EXCEEDED
04A1 D2A01A         JNC  SNERR      ;ERROR IF MORE THAN 8 OPEN FOR/N
04A4 328221         STA  FORNE      ;PUT IN TABLE
04A7 72             MOV  M,D        ;STORE IT
04A8 23             INX  H          ;POINT NEXT
04A9 73             MOV  M,E        ;STORE IT TOO
04AA 23             INX  H          ;POINT NEXT
04AB E5             PUSH H          ;SAVE H,L
04AC 212020         LXI  H,TVAR2    ;POINT STEP VALUE
04AF EF             RST  5          ;LOAD IT
04B0 E1             POP  H          ;RESTORE H,L
04B1 F7             RST  6          ;STORE IN STACK
04B2 E5             PUSH H          ;SAVE H,L
04B3 211C20         LXI  H,TVAR1    ;POINT 'TO' VALUE
04B6 EF             RST  5          ;LOAD IT
```

```
ADDR OBJECT   LINE TAG    OPCD OPRNDS       COMMENTS

04B7 E1       0823        POP  H            ;RESTORE H,L
04B8 F7       0824        RST  6            ;STORE IN STACK
04B9 EB       0825        XCHG              ;FLIP/FLOP
04BA 2A6321   0826        LHLD ENDLI        ;GET END ADDR
04BD 2B       0827        DCX  H            ;POINT ONE PRIOR
04BE EB       0828        XCHG              ;FLIP BACK
04BF 72       0829        MOV  M,D          ;STORE IT
04C0 23       0830        INX  H            ;POINT NEXT
04C1 73       0831        MOV  M,E          ;STORE IT
04C2 23       0832        INX  H            ;POINT NEXT
04C3 3A6221   0833        LDA  STMT+1       ;GET HIGH STMT ADDR
04C6 77       0834        MOV  M,A          ;PUT IT
04C7 23       0835        INX  H            ;POINT NEXT
04C8 3A6121   0836        LDA  STMT         ;GET LOW STMT ADDR
04CB 77       0837        MOV  M,A          ;PUT IT
04CC C3F401   0838        JMP  RUN          ;CONTINUE
04CF 2A6121   0839 FOR9:  LHLD STMT         ;GET ADDRESS OF STATEMENT
04D2 5E       0840        MOV  E,M          ;GET LENGTH CODE
04D3 1600     0841        MVI  D,0          ;INIT INCREMENT
04D5 19       0842        DAD  D            ;COMPUTE ADDR OF NEXT STATEMENT
04D6 7E       0843        MOV  A,M          ;GET NEW LEN CODE
04D7 B7       0844        ORA  A            ;SEE IF END OF PGM
04D8 CAAC1A   0845        JZ   NXERR        ;BRIF IT IS
04DB 226121   0846        SHLD STMT         ;SAVE ADDRESS
04DE FF       0847        RST  7            ;ADJUST H,L
04DF 03       0848        DB   3
04E0 CF       0849        RST  1            ;SKIP SPACES
04E1 11451F   0850        LXI  D,NEXTL      ;POINT 'NEXT'
04E4 D7       0851        RST  2            ;SEE IF IT IS A NEXT STMT
04E5 C2CF04   0852        JNZ  FOR9         ;LOOP IF NOT
04E8 CF       0853        RST  1            ;SKIP SPACES
04E9 3A1920   0854        LDA  INDX+1       ;GET FIRST CHAR
04EC BE       0855        CMP  M            ;COMPARE
04ED C2CF04   0856        JNZ  FOR9         ;BRIF NOT MATCH NEXT
04F0 3A1820   0857        LDA  INDX         ;GET 2ND CHAR
04F3 23       0858        INX  H            ;DITTO
04F4 FE20     0859        CPI  ' '          ;SEE IF SINGLE CHAR
04F6 CAFD04   0860        JZ   FORA         ;BRIF IT IS
04F9 BE       0861        CMP  M            ;COMPARE THE TWO
04FA C2CF04   0862        JNZ  FOR9         ;BRIF NOT EQUAL
04FD CF       0863 FORA:  RST  1            ;SKIP TO END (HOPEFULLY)
04FE 7E       0864        MOV  A,M          ;GET THE NON BLANK
04FF B7       0865        ORA  A            ;SEE IF END
0500 C2CF04   0866        JNZ  FOR9         ;BRIF NOT END
0503 C3F401   0867        JMP  RUN          ;ELSE, GO NEXT STMT
```

```
ADDR OBJECT  TAG    OPCD OPRNDS          COMMENTS

0506         IF:    EQU  $
             ;
             ;
             ;  STMT:  IF EXPR RELATION EXPR THEN STMT#
             ;
             ;
0506 CD0714         CALL EXPR            ;GO EVALUATE LEFT EXPR
0509 E5             PUSH H               ;SAVE H,L
050A 3AF521         LDA  NS              ;GET TYPE CODE
050D 321B20         STA  IFTYP           ;SAVE IT
0510 FEE7           CPI  231             ;TEST IF STRING
0512 C22405         JNZ  IF1             ;BRIF NOT
0515 210520         LXI  H,IOBUF         ;POINT BUFFER
0518 115320         LXI  D,STRIN         ;POINT RESULT
051B 1A             LDAX D               ;GET LEN
051C 3C             INR  A               ;PLUS ONE
051D 47             MOV  B,A             ;SAVE IT
051E CD241C         CALL COPYD           ;GO MOVE IT
0521 C32805         JMP  IF2             ;GO AROUND
0524 211C20  IF1:   LXI  H,TVAR1         ;GET ADDR OF TEMP STORAGE
0527 F7             RST  6               ;SAVE IT
0528 E1      IF2:   POP  H               ;RESTORE H,L
0529 AF             XRA  A               ;CLEAR A
052A 4F             MOV  C,A             ;SAVE IN REG C
052B 47             MOV  B,A             ;INIT REG
052C 7E      IF3:   MOV  A,M             ;GET OPERATOR
052D 04             INR  B               ;COUNT
052E FE3D           CPI  '='             ;TEST FOR EQUAL
0530 C23505         JNZ  IF4             ;BRIF IT IS
0533 0C             INR  C               ;ADD 1 TO C
0534 23             INX  H               ;POINT NEXT
0535 FE3E    IF4:   CPI  '>'             ;TEST FOR GREATER THAN
0537 C23D05         JNZ  IF5             ;BRIF IT IS
053A 0C             INR  C               ;ADD TWO
053B 0C             INR  C               ;TO REL CODE
053C 23             INX  H               ;POINT NEXT
053D FE3C    IF5:   CPI  '<'             ;TEST FOR LESS THAN
053F C24705         JNZ  IF6             ;BRIF IT IS
0542 79             MOV  A,C             ;GET REL CODE
0543 C604           ADI  4               ;PLUS FOUR
0545 4F             MOV  C,A             ;PUT BACK
0546 23             INX  H               ;POINT NEXT
0547 79      IF6:   MOV  A,C             ;GET REL CODE
0548 B7             ORA  A               ;TEST IT
0549 C5             PUSH B               ;SAVE B,C
054A CAA01A         JZ   SNERR           ;BRIF SOME ERROR
054D C1             POP  B               ;RESTORE B,C
054E 321A20         STA  REL             ;SAVE CODE
0551 78             MOV  A,B             ;GET COUNT
0552 FE02           CPI  2               ;TEST FOR TWO
0554 C22C05         JNZ  IF3             ;SEE IF MULTIPLE RELATION
```

```
ADDR OBJECT  TAG    OPCD OPRNDS       COMMENTS

0557 CD0714         CALL EXPR         ;GO EVALUATE RIGHT SIDE
055A 221220  IF7:   SHLD ADDR1        ;SAVE LOCATION OF THEN (IF ANY)
055D 3AF521         LDA  NS           ;GET TYPE CODE
0560 211B20         LXI  H,IFTYP      ;POINT LEFT TYPE
0563 BE             CMP  M            ;COMPARE
0564 C2A01A         JNZ  SNERR        ;BRIF MIXED
0567 FEE7           CPI  231          ;TEST IF STRING
0569 CAC505         JZ   IFF          ;BRIF IS
056C 211C20         LXI  H,TVAR1      ;POINT LEFT
056F CD530E         CALL FSUB         ;SUBTRACT LEFT FROM RIGHT
0572 3A1A20         LDA  REL          ;GET RELATION
0575 1F             RAR               ;TEST BIT 00
0576 D27F05         JNC  IF8          ;BRIF NO EQUAL TEST
0579 CD750F         CALL FTEST        ;GET STATUS OF FACC
057C CA9E05         JZ   TRUE         ;BRIF LEFT=RIGHT
057F 3A1A20  IF8:   LDA  REL          ;LOAD RELATION
0582 E602           ANI  2            ;MASK IT
0584 CA8D05         JZ   IF9          ;BRIF NO >
0587 CD750F         CALL FTEST        ;GET STATUS OF FACC
058A FA9E05         JM   TRUE         ;BRIF GT
058D 3A1A20  IF9:   LDA  REL          ;LOAD RELATION
0590 E604           ANI  4            ;MASK IT
0592 CAF401         JZ   FALSE        ;BRIF NO <
0595 CD750F         CALL FTEST        ;GET STATUS OF FACC
0598 FAF401         JM   FALSE        ;BRIF GT
059B CAF401         JZ   FALSE        ;BRIF ZERO (NOT EQUAL)
059E 2A1220  TRUE:  LHLD ADDR1        ;GET POINTER TO STATEMENT
05A1 116D1F         LXI  D,GOTOL      ;POINT 'GO TO'
05A4 D7             RST  2            ;GO COMPARE
05A5 CAE302         JZ   GOTO         ;BRIF IF ... GOTO NN
05A8 2A1220         LHLD ADDR1        ;GET POINTER TO STATEMENT
05AB 114C1F         LXI  D,GOSBL      ;POINT LITERAL
05AE D7             RST  2.           ;GO COMPARE
05AF CA4703         JZ   GOSUB        ;BRIF IF ... GOSUB NN
05B2 2A1220         LHLD ADDR1        ;GET POINTER TO STATEMENT
05B5 11361E         LXI  D,THENL      ;GET ADDR 'THEN'
05B8 D7             RST  2            ;GO COMPARE
05B9 C2A01A         JNZ  SNERR        ;BRIF ERROR
05BC CD351B  IFB:   CALL NUMER        ;TEST IF NUMERIC
05BF CAE302         JZ   GOTO         ;BRIF IT IS
05C2 C32102         JMP  RUN4         ;ELSE, MAY BE ANY STMT
01F4         FALSE: EQU  RUN
05C5 210520  IFF:   LXI  H,IOBUF      ;POINT PRIOR
05C8 46             MOV  B,M          ;GET LEN
05C9 115820         LXI  D,STRIN      ;POINT THIS
05CC 1A             LDAX D            ;GET LEN
05CD 4F             MOV  C,A          ;SAVE IT
05CE 13      IFG:   INX  D            ;POINT NEXT
05CF 23             INX  H            ;DITTO
05D0 78             MOV  A,B          ;GET LEFT LEN
05D1 B7             ORA  A            ;TEST IT
```



```
ADDR OBJECT  TAG    OPCD OPRNDS       COMMENTS

05D2 C2D705         JNZ  IFH          ;BRIF NOT ZERO
05D5 3620           MVI  M,' '        ;EXTEND WITH SPACE
05D7 79      IFH:   MOV  A,C          ;GET RIGHT LEN
05D8 B7             ORA  A            ;TEST IT
05D9 C2DF05         JNZ  IFI          ;BRIF NOT ZERO
05DC 3E20           MVI  A,' '        ;GET SPACE
05DE 12             STAX D            ;EXTEND
05DF 1A      IFI:   LDAX D            ;GET RIGHT CHAR
05E0 BE             CMP  M            ;TEST WITH LEFT
05E1 DA0406         JC   IFM          ;BRIF LEFT>RIGHT
05E4 C20906         JNZ  IFN          ;BRIF LEFT<RIGHT
05E7 78             MOV  A,B          ;GET LEFT COUNT
05E8 3D             DCR  A            ;SUBT ONE
05E9 FAED05         JM   IFJ          ;BRIF WAS ZERO
05EC 47             MOV  B,A          ;UPDATE CTR
05ED 79      IFJ:   MOV  A,C          ;GET RIGHT LEN
05EE 3D             DCR  A            ;SUBT ONE
05EF FAF305         JM   IFK          ;BRIF WAS ZERO
05F2 4F             MOV  C,A          ;UPDT CTR
05F3 78      IFK:   MOV  A,B          ;GET LEFT LEN
05F4 B1             ORA  C            ;COMPARE TO RIGHT
05F5 C2CE05         JNZ  IFG          ;BRIF BOTH NOT ZERO
05F8 0601           MVI  B,1          ;SET SW= EQUAL
05FA 3A1A20  IFL:   LDA  REL          ;GET RELATION
05FD A0             ANA  B            ;AND WITH RESULT
05FE CAF401         JZ   FALSE        ;BRIF FALSE
0601 C39E05         JMP  TRUE         ;ELSE, TRUE
0604 0602    IFM:   MVI  B,2          ;SET CODE
0606 C3FA05         JMP  IFL          ;JUMP
0609 0604    IFN:   MVI  B,4          ;SET CODE
060B C3FA05         JMP  IFL          ;JUMP
```

```
ADDR OBJECT  TAG     OPCD OPRNDS          COMMENTS

060E         LET:    EQU  $
             ;
             ;
             ;  STMT:  [LET] VAR = EXPR
             ;
             ;
060E CD4008          CALL GETS8           ;GO GET ADDRESS OF VARIABLE
0611 C5              PUSH B               ;SAVE NAME
0612 D5              PUSH D               ;SAVE ADDRESS
0613 7E              MOV  A,M             ;LOAD CHAR
0614 FE3D            CPI  '='             ;TEST FOR EQUAL SIGN
0616 CA2906          JZ   LET1            ;BRIF IS
0619 3A6921          LDA  EDSW            ;GET MODE SW
061C B7              ORA  A               ;TEST IT
061D CAA01A          JZ   SNERR           ;BRIF LET ERROR
0620 21641D          LXI  H,WHATL         ;POINT LITERAL
0623 CD9119          CALL TERMM           ;GO PRINT IT
0626 C3B600          JMP  GETCM           ;GO TO COMMAND
0629 23      LET1:   INX  H               ;POINT NEXT
062A CD0714          CALL EXPR            ;GO EVALUATE EXPRESSION
062D CD681A          CALL EOL             ;ERROR IF NOT END-OF-LINE
0630 E1              POP  H               ;RESTORE ADDRESSS
0631 D1              POP  D               ;RESTORE NAME
0632 7B              MOV  A,E             ;GET TYPE
0633 B7              ORA  A               ;TEST IT
0634 3AF521          LDA  NS              ;GET RESULT TYPE
0637 FA4306          JM   LET2            ;BRIF STRING
063A FEE3            CPI  227             ;TEST IF NUMERIC
063C C2A01A          JNZ  SNERR           ;BRIF MIXED MODE
063F F7              RST  6               ;GO STORE VARIABLE
0640 C3F401          JMP  RUN             ;CONTINUE
0643 FEE7    LET2:   CPI  231             ;TEST IF STRING
0645 C2A01A          JNZ  SNERR           ;BRIF MIXED MODE
0648 CD4E06          CALL LET2A           ;GO STORE IT
064B C3F401          JMP  RUN             ;CONTINUE
             ;
064E 115820  LET2A:  LXI  D,STRIN         ;POINT STRING BUFFER
0651 1A              LDAX D               ;GET NEW LEN
0652 96              SUB  M               ;MINUS OLD LEN
0653 CAA306          JZ   LET8            ;BRIF SAME LENGTH
0656 54              MOV  D,H             ;COPY H,L
0657 5D              MOV  E,L             ;TO D,E
0658 7E              MOV  A,M             ;GET LEN
0659 3C              INR  A               ;TRUE LEN
065A 13      LET3:   INX  D               ;POINT NEXT
065B 3D              DCR  A               ;DECR CTR
065C C25A06          JNZ  LET3            ;LOOP
065F 13              INX  D               ;SKIP
0660 13              INX  D               ;AGAIN
0661 1A              LDAX D               ;GET LO NAM
0662 4F              MOV  C,A             ;SAVE
```

```
ADDR OBJECT   TAG     OPCD OPRNDS       COMMENTS

0663 13               INX  D            ;GET HI NAME
0664 1A               LDAX D            ;LOAD IT
0665 47               MOV  B,A          ;SAVE
0666 C5               PUSH B            ;SAVE NAME
0667 2B               DCX  H            ;POINT NEXT ENTRY
0668 7E       LET4:   MOV  A,M          ;GET NEXT
0669 B7               ORA  A            ;TEST IF END
066A CA8106           JZ   LET6         ;BRIF IS
066D E5               PUSH H            ;SAVE H,L
066E 2B               DCX  H            ;SKIP NEXT
066F 2B               DCX  H            ;POINT LEN
0670 46               MOV  B,M          ;GET HI LEN
0671 2B               DCX  H            ;POINT LO
0672 4E               MOV  C,M          ;GET LO LEN
0673 E1               POP  H            ;RESTORE H,L
0674 7E       LET5:   MOV  A,M          ;GET A BYTE
0675 12               STAX D            ;COPY
0676 2B               DCX  H            ;POINT NEXT
0677 1B               DCX  D            ;DITTO
0678 03               INX  B            ;ADD TO CTR
0679 78               MOV  A,B          ;GET HI
067A B1               ORA  C            ;TEST IF ZERO
067B C27406           JNZ  LET5         ;LOOP
067E C36806           JMP  LET4         ;CONTINUE
0681 EB       LET6:   XCHG              ;PUT NEW ADDR TO H,L
0682 C1               POP  B            ;GET NAME
0683 70               MOV  M,B          ;STORE HI BYTE
0684 2B               DCX  H            ;POINT NEXT
0685 71               MOV  M,C          ;STORE LO
0686 115820           LXI  D,STRIN      ;GET NEW LEN
0689 1A               LDAX D            ;LOAD IT
068A 06FF             MVI  B,255        ;INIT HI COMPLEMENT
068C C605             ADI  5            ;COMPUTE ENTRY LENGTH
068E CA9606           JZ   LET7         ;BRIF 256 BYTES
0691 D29606           JNC  LET7         ;BRIF LESS 256
0694 06FE             MVI  B,254        ;SET BIT OFF
0696 2F       LET7:   CMA               ;1'S COMPLEMENT
0697 3C               INR  A            ;THEN 2'S
0698 4F               MOV  C,A          ;SAVE LO LEN
0699 2B               DCX  H            ;POINT NEXT
069A 70               MOV  M,B          ;STORE HI LEN
069B 2B               DCX  H            ;POINT NEXT
069C 71               MOV  M,C          ;STORE LO LEN
069D FF               RST  7            ;ADJUST H,L
069E 03               DB   3
069F 09               DAD  B            ;COMPUTE END OF ENTRY
06A0 3600             MVI  M,0          ;MARK NEW END
06A2 23               INX  H            ;POINT 1ST BYTE
06A3 1A       LET8:   LDAX D            ;GET LEN
06A4 3C               INR  A            ;TRUE LEN
06A5 47               MOV  B,A          ;SAVE LEN
```



```
ADDR OBJECT   TAG     OPCD OPRNDS       COMMENTS

06A6 1A       LET9:   LDAX D            ;GET A BYTE
06A7 77               MOV  M,A          ;COPY IT
06A8 23               INX  H            ;POINT NEXT
06A9 13               INX  D            ;DITTO
06AA 05               DCR  B            ;SUBT CTR
06AB C2A606           JNZ  LET9         ;LOOP
06AE C9               RET               ;RETURN
```

```
ADDR OBJECT  TAG      OPCD OPRNDS       COMMENTS

06AF         NEXT:    EQU  $
             ;
             ;
             ; STMT:   NEXT VAR
             ;
             ;
06AF CDD41B           CALL VAR          ;GET VARIABLE NAME
06B2 CD681A           CALL EOL          ;ERROR IF NOT END-OF-LNE
06B5 EB               XCHG              ;FLIP/FLOP
06B6 221320           SHLD INDX         ;SAVE VAR NAME
06B9 E5               PUSH H            ;SAVE VAR NAME
06BA 218221           LXI  H,FORNE      ;POINT FOR/NEXT TABLE
06BD 46               MOV  B,M          ;GET SIZE
06BE 78               MOV  A,B          ;LOAD IT
06BF B7               ORA  A            ;TEST IT
06C0 CAAC1A           JZ   NXERR        ;BRIF TABLE EMPTY
06C3 23               INX  H            ;POINT NEXT
06C4 D1               POP  D            ;RESTORE VAR NAME
06C5 7E      NEXT1:   MOV  A,M          ;GET 1ST BYTE
06C6 23               INX  H            ;POINT NEXT
06C7 BA               CMP  D            ;COMPARE
06C8 C2D006           JNZ  NEXT2        ;BRIF NOT EQUAL
06CB 7E               MOV  A,M          ;GET 2ND BYTE
06CC BB               CMP  E            ;COMPARE
06CD CAD906           JZ   NEXT3        ;BRIF EQUAL
06D0 FF      NEXT2:   RST  7            ;ADJUST H,L
06D1 0D               DB   13
06D2 05               DCR  B            ;DECR COUNT
06D3 C2C506           JNZ  NEXT1        ;LOOP
06D6 C3AC1A           JMP  NXERR        ;GO PUT ERROR MSG
06D9 3A8221  NEXT3:   LDA  FORNE        ;GET ORIG COUNT
06DC 90               SUB  B            ;MINUS REMAIN
06DD 3C               INR  A            ;PLUS ONE
06DE 328221           STA  FORNE        ;STORE NEW COUNT
06E1 23               INX  H            ;POINT ADDR
06E2 E5               PUSH H            ;SAVE H,L ADDR
06E3 CD3F1B           CALL SEARC        ;GO GET ADDR OF INDEX
06E6 EB               XCHG              ;PUT TO H,L
06E7 221220           SHLD ADDR1        ;SAVR IT
06EA EF               RST  5            ;LOAD INDEX
06EB E1               POP  H            ;GET H,L (TBL)
06EC E5               PUSH H            ;RE-SAVE
06ED CD590D           CALL FADD         ;ADD STEP VALUE
06F0 211C20           LXI  H,TVAR1      ;POINT TEMP AREA
06F3 F7               RST  6            ;SAVE NEW INDEX
06F4 E1               POP  H            ;GET H,L (TBL)
06F5 E5               PUSH H            ;RE-SAVE
06F6 FF               RST  7            ;GET LEN TO NEXT
06F7 04               DB   4
06F8 CD530E           CALL FSUB         ;SUBTRACT TO VALUE
06FB CA1807           JZ   NEXT6        ;BRIF ZERO
```



```
ADDR OBJECT  TAG      OPCD OPRNDS       COMMENTS

06FE E1               POP  H            ;GET H,L (PTR TO STEP)
06FF E5               PUSH H            ;RE-SAVE
0700 7E               MOV  A,M          ;GET SIGN&EXPONENT OF STEP
0701 B7               ORA  A            ;TEST IT
0702 3A4820           LDA  FACC         ;GET SIGN & EXPON OF DIFF
0705 FA1407           JM   NEXT5        ;BRIF NEGATIVE
0708 B7      NEXT4:   ORA  A            ;TEST SIGN OF DIFF
0709 FA1807           JM   NEXT6        ;BRIF LESS THAN TO-EXPR
070C 218221  NEXT7:   LXI  H,FORNE      ;GET ADDR TABLE
070F 35               DCR  M            ;SUBTRACT ONE FROM COUNT
0710 D1               POP  D            ;ADJUST STACK
0711 C3F401           JMP  RUN          ;GO STMT AFTER NEXT
0714 B7      NEXT5:   ORA  A            ;TEST SIGN OF DIFFERENCE
0715 FA0C07           JM   NEXT7        ;BRIF END OF LOOP
0718 E1      NEXT6:   POP  H            ;GET PTR TO TBL
0719 FF               RST  7            ;ADJUST H,L
071A 08               DB   8
071B 56               MOV  D,M          ;GET HI BYTE
071C 23               INX  H            ;POINT NEXT
071D 5E               MOV  E,M          ;GET LO BYTE
071E 23               INX  H            ;POINT NEXT
071F 7E               MOV  A,M          ;GET HI BYTE
0720 326221           STA  STMT+1       ;SAVE
0723 23               INX  H            ;POINT NEXT
0724 7E               MOV  A,M          ;GET LO BYTE
0725 326121           STA  STMT         ;SAVE
0728 EB               XCHG              ;H,L = ADDR OF STMT AFTR FOR
0729 CD681A           CALL EOL          ;SETUP MULTI PTR
072C 2A6121           LHLD STMT         ;GET ADDR OF FOR STMT
072F 23               INX  H            ;POINT LINE NUM
0730 227E21           SHLD LINE         ;SAVE ADDR LINE
0733 211C20           LXI  H,TVAR1      ;POINT UPDTED VALUE
0736 EF               RST  5            ;GO LOAD IT
0737 2A1220           LHLD ADDR1        ;GET ADDR OF INDEX
073A F7               RST  6            ;GO STORE IT
073B C3F401           JMP  RUN          ;CONTINUE WITH STMT AFTER FOR
```

```
ADDR OBJECT   TAG     OPCD OPRNDS          COMMENTS

073E          INPUT:  EQU  $
              ;
              ;
              ;  STMT:   INPUT VAR [, VAR, VAR]
              ;
              ;
073E 11691D           LXI  D,LLINE         ;POINT 'LINE'
0741 E5               PUSH H               ;SAVE H,L ADDR
0742 D7               RST  2               ;GO COMPARE
0743 CAC207           JZ   INPL            ;BRIF EQUAL
0746 D1               POP  D               ;ELSE, RESTORE H,L ADDR
0747 210520           LXI  H,IOBUF         ;GET ADDR OF BUFFER
074A 221220           SHLD ADDR1           ;SAVE ADDR
074D 3600             MVI  M,0             ;MARK BUFFER EMPTY
074F EB               XCHG                 ;FLIP/BACK
0750 CF       INPU1:  RST  1               ;SKIP SPACES
0751 FE27             CPI  39              ;TEST IF QUOTE
0753 CA5B07           JZ   INPU2           ;BRIF IS
0756 FE22             CPI  '"'             ;TEST IF INPUT LITERAL
0758 C27E07           JNZ  INPU6           ;BRIF NOT
075B 4F       INPU2:  MOV  C,A             ;SAVE DELIM
075C 110520           LXI  D,IOBUF         ;POINT BUFFER
075F 23       INPU3:  INX  H               ;POINT NEXT
0760 7E               MOV  A,M             ;LOAD IT
0761 B9               CMP  C               ;TEST IF END
0762 CA6A07           JZ   INPU4           ;BRIF IS
0765 12               STAX D               ;PUT TO BUFF
0766 13               INX  D               ;POINT NEXT
0767 C35F07           JMP  INPU3           ;LOOP
076A 23       INPU4:  INX  H               ;SKIP TRAILING QUOTE
076B EB               XCHG                 ;PUT ADDR TO H,L
076C 36FE             MVI  M,254           ;MARK END
076E CD8919           CALL TERMO           ;GO PRINT PROMPT
0771 EB               XCHG                 ;GET H,L
0772 CF               RST  1               ;SKIP TO NON BLANK
0773 FE2C             CPI  ','             ;TEST IF COMMA
0775 CA7D07           JZ   INPU5           ;BRIF IS
0778 FE3B             CPI  ';'             ;TEST IF COMMA
077A C27E07           JNZ  INPU6           ;BRIF NOT
077D 23       INPU5:  INX  H               ;SKIP IT
077E CD4008   INPU6:  CALL GETS8           ;GO GET VAR ADDR
0781 E5               PUSH H               ;SAVE H ADDR
0782 D5               PUSH D               ;SAVE VAR ADDR
0783 2A1220           LHLD ADDR1           ;GET ADDR PREV BUFFER
0786 7E               MOV  A,M             ;LOAD CHAR
0787 FE2C             CPI  ','             ;TEST IF COMMA
0789 23               INX  H               ;POINT NEXT
078A CA9207           JZ   INPU7           ;BRIF CONTINUE FROM PREV
078D 3E3F             MVI  A,'?'           ;PROMPT CHAR
078F CDD518           CALL TERMI           ;GO READ FROM TTY
0792 CF       INPU7:  RST  1               ;SKIP SPACES
```



```
ADDR OBJECT   TAG     OPCD OPRNDS          COMMENTS

0793 79               MOV  A,C             ;GET LO NAME
0794 B7               ORA  A               ;TEST IT
0795 FAB907           JM   INPUA           ;BRIF STRING
0798 CD1B0B           CALL FIN             ;GO CONVERT TO FLOATING
079B CF               RST  1               ;SKIP SPACES
079C FE2C             CPI  ','             ;TEST IF COMMA
079E CAA507           JZ   INPU8           ;BRIF IS
07A1 B7               ORA  A               ;TEST IF END OF LINE
07A2 C2B01A           JNZ  CVERR           ;BRIF ERROR
07A5 221220   INPU8:  SHLD ADDR1           ;SAVE ADDRESS
07A8 E1               POP  H               ;GET VAR ADDR
07A9 F7               RST  6               ;GO STORE THE NUMBER
07AA E1       INPU9:  POP  H               ;RESTORE STMT POINTER
07AB 7E               MOV  A,M             ;GET CHAR
07AC FE2C             CPI  ','             ;TEST FOR COMMA
07AE 23               INX  H               ;POINT NEXT
07AF CA5007           JZ   INPU1           ;RECURSIVE IF COMMA
07B2 2B               DCX  H               ;POINT BACK
07B3 CD681A   INPUB:  CALL EOL             ;ERROR IF NOT END OF LINE
07B6 C3F401           JMP  RUN             ;CONTINUE NEXT STMT
07B9 CDFE07   INPUA:  CALL GETST           ;GO GET THE STRING
07BC 221220           SHLD ADDR1           ;SAVE ADDRESS
07BF C3AA07           JMP  INPU9           ;CONTINUE
```

```
ADDR OBJECT   LINE TAG    OPCD OPRNDS          COMMENTS

07C2               INPL:  EQU  $
                   ;
                   ;
                   ; STMT  INPUT LINE A$
                   ;
                   ;
07C2 D1                   POP  D               ;DUMMY POP TO ADJUST STACK
07C3 CDD41B               CALL VAR             ;GET STRING NAME
07C6 7B                   MOV  A,E             ;LOAD LO BYTE
07C7 B7                   ORA  A               ;TEST IT
07C8 F2A01A               JP   SNERR           ;BRIF NOT STRING VARIABLE
07CB CD3F1B               CALL SEARC           ;ELSE, GET ADDRESS
07CE D5                   PUSH D               ;SAVE ON STACK
07CF CD681A               CALL EOL             ;ERROR IF NOT END-OF-LINE
07D2 3E01                 MVI  A,1             ;GET ON SETTING
07D4 326021               STA  ILSW            ;SET INPUT LINE SWITCH
07D7 3E3F                 MVI  A,'?'           ;SET PROMPT = QM
07D9 CDD518               CALL TERMI           ;GO READ A LINE
07DC 0600                 MVI  B,0             ;INIT COUNT
07DE 115920               LXI  D,STRIN+1       ;POINT STRING BUFFER
07E1 210520               LXI  H,IOBUF         ;POINT INPUT BUFFER
07E4 7E            INPL1: MOV  A,M             ;GET NEXT BYTE
07E5 B7                   ORA  A               ;TEST IT
07E6 CAF007               JZ   INPL2           ;BRIF END
07E9 04                   INR  B               ;ADD TO COUNT
07EA 12                   STAX D               ;PUT TO STRING BUFF
07EB 13                   INX  D               ;POINT NEXT
07EC 23                   INX  H               ;DITTO
07ED C3E407               JMP  INPL1           ;LOOP
07F0 326021        INPL2: STA  ILSW            ;RESET SWITCH
07F3 78                   MOV  A,B             ;GET COUNT
07F4 325820               STA  STRIN           ;SET STRING LENGTH
07F7 E1                   POP  H               ;GET ADDRESS OF VARIABLE
07F8 CD4E06               CALL LET2A           ;GO STORE THE STRING
07FB C3F401               JMP  RUN             ;GO NEXT STMT
```



```
ADDR OBJECT   LINE TAG    OPCD OPRNDS          COMMENTS

07FE 115820        GETST: LXI  D,STRIN         ;POINT BUFFER
0801 0600                 MVI  B,0             ;INIT CTR
0803 7E                   MOV  A,M             ;GET THE CHAR
0804 FE22                 CPI  '"'             ;TEST IF LIT TYPE
0806 CA1F08               JZ   GETS2           ;BRIF IS
0809 FE27                 CPI  39              ;TEST IF QUOTED LITERAL
080B CA1F08               JZ   GETS2           ;BRIF IS
080E FE2C          GETS1: CPI  ','             ;TEST IF COMMA
0810 CA3208               JZ   GETS5           ;BRIF IS
0813 B7                   ORA  A               ;TEST IF END
0814 CA3208               JZ   GETS5           ;BRIF IS
0817 04                   INR  B               ;COUNT IT
0818 13                   INX  D               ;POINT NEXT
0819 12                   STAX D               ;PUT CHAR
081A 23                   INX  H               ;POINT NEXT
081B CF                   RST  1               ;SKIP SPACES
081C C30E08               JMP  GETS1           ;LOOP
081F 4F            GETS2: MOV  C,A             ;SAVE DELIM
0820 23            GETS3: INX  H               ;SKIP THE QUOTE
0821 7E                   MOV  A,M             ;GET NEXT CHAR
0822 B9                   CMP  C               ;TEST IF END OF LITERAL
0823 CA3008               JZ   GETS4           ;BRIF IS
0826 B7                   ORA  A               ;TEST IF END LINE
0827 CAB01A               JZ   CVERR           ;BRIF IS
082A 04                   INR  B               ;COUNT IT
082B 13                   INX  D               ;POINT NEXT
082C 12                   STAX D               ;PUT CHAR
082D C32008               JMP  GETS3           ;LOOP
0830 23            GETS4: INX  H               ;SKIP END QUOTE
0831 CF                   RST  1               ;SKIP TRAILING SPACES
0832 115820        GETS5: LXI  D,STRIN         ;POINT BEGIN BUFFER
0835 78                   MOV  A,B             ;GET COUNT
0836 12                   STAX D               ;PUT COUNT
0837 D1                   POP  D               ;GET RETURN ADDR
0838 EB                   XCHG                 ;FLIP/FLOP
0839 E3                   XTHL                 ;PUT RET ON STACK, H,L OF VAR IN
083A D5                   PUSH D               ;SAVE H,L OF LOC
083B CD4E06               CALL LET2A           ;GO STORE STRING
083E E1                   POP  H               ;RESTORE LOCATION
083F C9                   RET                  ;RETURN
0840 CDD41B        GETS8: CALL VAR             ;GET VAR NAME
0843 D5                   PUSH D               ;SAVE ON STACK
0844 7A                   MOV  A,D             ;GET HI BYTE
0845 B7                   ORA  A               ;TEST IF ARRAY
0846 F25D08               JP   GETS9           ;BRIF NOT
0849 CD3F1B               CALL SEARC           ;GO GET ARRAY PARAMS
084C 3EFF                 MVI  A,255           ;TURN ON SW
084E 326321               STA  DIMSW           ;SET IT
0851 E3                   XTHL                 ;SWAP ADDR ON STACK
0852 CD0714               CALL EXPR            ;GO GET ROW, COL PTRS
0855 E3                   XTHL                 ;SWAP ADDR ON STACK
```

```
ADDR OBJECT   LINE TAG     OPCD OPRNDS            COMMENTS

0856 CDD20A                CALL SUBSC             ;GO POINT TO ENTRY
0859 EB                    XCHG                   ;EXCHANGE
085A E1                    POP  H                 ;GET ADDRESS OF STMT
085B C1                    POP  B                 ;GET NAME
085C C9                    RET                    ;RETURN
085D CD3F1B        GETS9:  CALL SEARC             ;FIND ADDR
0860 C1                    POP  B                 ;RESTORE NAME
0861 C9                    RET                    ;RETURN
```



```
ADDR OBJECT   LINE TAG     OPCD OPRNDS            COMMENTS

0862               READ:   EQU  $
                   ;
                   ;  STMT: READ VAR [,VAR ...]
                   ;
0862 CD4008                CALL GETS8             ;GET VAR ADDR
0865 E5                    PUSH H                 ;SAVE H,L
0866 D5                    PUSH D                 ;SAVE D,E
0867 2AF621                LHLD DATAP             ;GET DATA STMT POINTER
086A 7E                    MOV  A,M               ;LOAD THE CHAR
086B B7                    ORA  A                 ;TEST IF END OF STMT
086C C28B08                JNZ  READ2             ;BRIF NOT END OF STMT
086F 23                    INX  H                 ;POINT START NEXT STMT
0870 7E            READ1:  MOV  A,M               ;LOAD LEN
0871 22F621                SHLD DATAP             ;SAVE ADDR
0874 B7                    ORA  A                 ;TEST IF END OF PGM
0875 CAA81A                JZ   DAERR             ;BRIF OUT OF DATA
0878 FF                    RST  7                 ;ADJUST H,L
0879 03                    DB   3
087A 11331F                LXI  D,DATAL           ;POINT 'DATA'
087D D7                    RST  2                 ;COMPARE
087E CA8B08                JZ   READ2             ;BRIF IT IS DATA STMT
0881 2AF621                LHLD DATAP             ;GET ADDR START
0884 5E                    MOV  E,M               ;GET LEN CODE
0885 1600                  MVI  D,0               ;CLEAR D
0887 19                    DAD  D                 ;POINT NEXT STMT
0888 C37008                JMP  READ1             ;LOOP NEXT STMT
088B CF            READ2:  RST  1                 ;SKIP SPACES
088C 79                    MOV  A,C               ;LOAD LO NAME
088D B7                    ORA  A                 ;TEST IT
088E FAB308                JM   READ6             ;BRIF STRING
0891 CD1B08                CALL FIN               ;GO CONVERT VALUE
0894 7E                    MOV  A,M               ;GET CHAR WHICH STOPPED US
0895 FE2C                  CPI  ','               ;TEST IF COMMA
0897 C2AC08                JNZ  READ5             ;BRIF NOT
089A 23                    INX  H                 ;POINT NEXT
089B 22F621        READ3:  SHLD DATAP             ;SAVE ADDRESS
089E E1                    POP  H                 ;RESTORE ADDR OF VAR
089F F7                    RST  6                 ;STORE THE VALUE
08A0 E1            READ4:  POP  H                 ;RESTORE POINTER TO STM
08A1 7E                    MOV  A,M               ;GET THE CHAR
08A2 FE2C                  CPI  ','               ;TEST IF COMMA
08A4 23                    INX  H                 ;POINT NEXT
08A5 CA6208                JZ   READ              ;RECURSIVE IF IT IS
08A8 2B                    DCX  H                 ;RESET
08A9 C3B307                JMP  INPUB             ;CONTINUE
08AC B7            READ5:  ORA  A                 ;TEST IF END OF STMT
08AD CA9B08                JZ   READ3             ;BRIF OK
08B0 C3B01A                JMP  CVERR             ;GO PROCESS ERROR
08B3 CDFE07        READ6:  CALL GETST             ;GO GET STRING
08B6 7E                    MOV  A,M               ;GET CHAR
08B7 FE2C                  CPI  ','               ;TEST IF COMMA
```

```
ADDR OBJECT   LINE TAG     OPCD OPRNDS            COMMENTS

08B9 CAC308                JZ   READ7             ;BRIF IS
08BC B7                    ORA  A                 ;TEST IF END
08BD C2AC08                JNZ  READ5             ;BRIF NOT
08C0 C3C408                JMP  READ8             ;GO AROUND
08C3 23             READ7: INX  H                 ;POINT PAST
08C4 22F621         READ8: SHLD DATAP             ;SAVE ADDRESS
08C7 C3A008                JMP  READ4             ;CONTINUE
```



```
ADDR OBJECT   LINE TAG     OPCD OPRNDS            COMMENTS

08CA                OUT:   EQU  $
                    ;
                    ; STMT: OUT ADDR,VALUE
                    ;
                    ;
08CA CD0714                CALL EXPR              ;GO EVALUATE ADDRESS
08CD 7E                    MOV  A,M               ;GET DELIM
08CE FE2C                  CPI  ','               ;TEST IF COMMA
08D0 C2A01A                JNZ  SNERR             ;BRIF NOT
08D3 23                    INX  H                 ;SKIP OVER COMMA
08D4 CD3B1C                CALL FBIN              ;CONVERT TO BINARY IN A-REG
08D7 110F20                LXI  D,OUTA            ;POINT INSTR
08DA EB                    XCHG                   ;PUT TO H,L
08DB 36D3                  MVI  M,211             ;OUT INSTR
08DD 23                    INX  H                 ;POINT NEXT
08DE 77                    MOV  M,A               ;PUT ADDR
08DF 23                    INX  H                 ;POINT NEXT
08E0 36C9                  MVI  M,201             ;RET INSTR
08E2 EB                    XCHG                   ;RESTORE ORIG H,L
08E3 CD0714                CALL EXPR              ;GO EVAL DATA BYTE
08E6 CD681A                CALL EOL               ;ERROR IF NOT END OF STATMENT
08E9 CD3B1C                CALL FBIN              ;CONVERT TO BINARY
08EC CD0F20                CALL OUTA              ;GO PUT THE BYTE
08EF C3F401                JMP  RUN               ;GO NEXT STMT
```

```
ADDR OBJECT    LINE TAG     OPCD OPRNDS             COMMENTS

08F2           1456 END:    EQU  $
               1457 ;
               1458 ;
               1459 ;  STMT:  END
               1460 ;
               1461 ;
08F2 3A6621    1462         LDA  TAPES              ;GET PAPER TAPE SWITCH
08F5 FE02      1463         CPI  2                  ;TEST IF PUNCH COMMAND
08F7 C29800    1464         JNZ  RDY                ;BRIF NOT
08FA CDB41A    1465         CALL HDRTL              ;GO PUT TRAILER
08FD C39101    1466         JMP  KEY                ;TERMINATE
```



```
ADDR OBJECT    LINE TAG     OPCD OPRNDS             COMMENTS

0900           1468 STOP:   EQU  $
               1469 ;
               1470 ;  STMT:  STOP
               1471 ;
               1472 ;
0900 CD681A    1473         CALL EOL                ;POINT END OF LINE
0903 21D11E    1474         LXI  H,STOPM            ;POINT MESSAGE: "STOP AT LINE "
0906 CD9119    1475         CALL TERMM              ;GO WRITE IT
0909 CDFC1B    1476         CALL PRLIN              ;GO PRINT LINE NUMBER
090C 3A6721    1477         LDA  RUNSW              ;GET RUN TYPE
090F B7        1478         ORA  A                  ;TEST IT
0910 C29800    1479         JNZ  RDY                ;BRIF IMMED
0913 326521    1480         STA  MULTI              ;CLEAR MULTI SW
0916 2A6121    1481         LHLD STMT               ;GET ADDR OF PREV STMT
0919 5E        1482         MOV  E,M                ;GET LEN
091A 1600      1483         MVI  D,0                ;CLEAR HI BYTE
091C 19        1484         DAD  D                  ;POINT NEXT
091D 23        1485         INX  H                  ;POINT LINE NUMBER
091E 227E21    1486         SHLD LINE               ;SAVE ADDR
0921 110020    1487         LXI  D,LINEN            ;POINT AREA
0924 CDDD19    1488         CALL LINEO              ;GO CONVERT LINE NUMBER
0927 EB        1489         XCHG                    ;FLIP TO H,L
0928 3600      1490         MVI  M,0                ;MARK END
092A C39800    1491         JMP  RDY                ;GO TO READY MSG
```

```
ADDR OBJECT     LINE TAG      OPCD OPRNDS            COMMENTS

092D            1493 RANDO:   EQU  $
                1494 ;
                1495 ;
                1496 ;  STMT:  RANDOMIZE
                1497 ;
                1498 ;
092D CD681A     1499          CALL EOL               ;ERROR IF NOT END-OF-LINE
0930 3E01       1500          MVI  A,1               ;LOAD A ONE
0932 327C21     1501          STA  RNDSW             ;SET SWITCH = TRUE RANDOM
0935 116C21     1502          LXI  D,TRNDX           ;POINT 'TRUE' RANDOM NUMBERS
0938 217421     1503          LXI  H,RNDX            ;POINT RECEIVE
093B 0608       1504          MVI  B,8               ;LOOP CTR
093D CD241C     1505          CALL COPYD             ;GO MOVE IT
0940 C3F401     1506          JMP  RUN               ;CONTINUE
```



```
ADDR OBJECT     LINE TAG      OPCD OPRNDS            COMMENTS

0943            1508 ON:      EQU  $
                1509 ;
                1510 ;
                1511 ;  STMT:  ON EXPR GOTO NNN NNNN NNNN
                1512 ;                    GOSUB
                1513 ;
                1514 ;
0943 CD0714     1515          CALL EXPR              ;GO EVALUATE EXPRESSION
0946 CD5B1C     1516          CALL FBIN              ;GET BINARY NUMBER IN ACC
0949 B7         1517          ORA  A                 ;TEST RESULT
094A CAA01A     1518          JZ   SNERR             ;BRIF ZERO (ERROR)
094D 4F         1519          MOV  C,A               ;SAVE VALUE
094E 0D         1520          DCR  C                 ;LESS ONE
094F AF         1521          XRA  A                 ;GET A ZERO
0950 321A20     1522          STA  REL               ;TURN OFF SWITCH
0953 11D01F     1523          LXI  D,GOTOL           ;POINT LITERAL
0956 E5         1524          PUSH H                 ;SAVE H,L ADDRESS
0957 D7         1525          RST  2                 ;GO COMPARE
0958 CA6909     1526          JZ   ON3               ;BRIF ON...GOTO
095B E1         1527          POP  H                 ;ELSE, RESTORE H,L
095C 114C1F     1528          LXI  D,GOSBL           ;POINT LITERAL
095F D7         1529          RST  2                 ;GO COMPARE
0960 C2A01A     1530          JNZ  SNERR             ;BRIF ERROR
0963 3E01       1531          MVI  A,1               ;GET ON SETTING
0965 321A20     1532          STA  REL               ;SET SWITCH
0968 E5         1533          PUSH H                 ;DUMMY PUSH
0969 D1         1534 ON3:     POP  D                 ;ADJUST STACK
096A 79         1535 ON3A:    MOV  A,C               ;GET COUNT
096B B7         1536          ORA  A                 ;TEST IT
096C CA8B09     1537          JZ   ON6               ;BRIF VALUE 1
096F CF         1538          RST  1                 ;ELSE, SKIP BLANKS
0970 B7         1539          ORA  A                 ;TEST IF END OF LINE
0971 CAA01A     1540          JZ   SNERR             ;BRIF IS
0974 FE2C       1541          CPI  ','               ;TEST IS COMMA
0976 C27D09     1542          JNZ  ON4               ;BRIF NOT
0979 23         1543          INX  H                 ;SKIP COMMA
097A C36A09     1544          JMP  ON3A              ;CONTINUE
097D CD351B     1545 ON4:     CALL NUMER             ;GO TEST IF NUMERIC
0980 C28709     1546          JNZ  ON5               ;BRIF NOT
0983 23         1547          INX  H                 ;POINT NEXT
0984 C37D09     1548          JMP  ON4               ;LOOP
0987 0D         1549 ON5:     DCR  C                 ;SUB ONE FROM COUNT
0988 C26A09     1550          JNZ  ON3A              ;LOOP TILL JUST BEFORE STMT#
098B CD8C1A     1551 ON6:     CALL NOTEO             ;ERROR IF NOT END-OF-LINE
098E FE2C       1552          CPI  ','               ;TEST IF COMMA
0990 C29709     1553          JNZ  ON7               ;BRIF NOT
0993 23         1554          INX  H                 ;POINT NEXT
0994 C38B09     1555          JMP  ON6               ;LOOP
0997 CD351B     1556 ON7:     CALL NUMER             ;TEST IF NUMERIC
099A C2A01A     1557          JNZ  SNERR             ;BRIF NOT
099D CDC01A     1558          CALL PACK              ;GET THE LINE NUMBER
```

```
ADDR OBJECT   LINE TAG     OPCD OPRNDS          COMMENTS

09A0 7E            ON8:    MOV  A,M             ;GET NEXT CHAR
09A1 CD311A                CALL TSTEL           ;TEST IF END STMT
09A4 CAAB09                JZ   ON9             ;BRIF END
09A7 23                    INX  H               ;POINT NEXT
09A8 C3A009                JMP  ON8             ;LOOP
09AB CD681A        ON9:    CALL EOL             ;SET END OF LINE POINTERS
09AE 3A1A20                LDA  REL             ;GET TYPE (GOTO OR GOSUB)
09B1 B7                    ORA  A               ;TEST IT
09B2 C25003                JNZ  GOSU1           ;BRIF GOSUB
09B5 214522                LXI  H,BEGPR         ;POINT START OF PROGRAM
09B8 C3FB02                JMP  GOTO2           ;BR TO GOTO LOOKUP
```



```
ADDR OBJECT   LINE TAG     OPCD OPRNDS          COMMENTS

09BB               CHANG:  EQU  $
                   ;
                   ;  STATEMENT:  CHANGE A$ TO X    - OR -
                   ;
                   ;              CHANGE X TO A$
                   ;
09BB CDD41B                CALL VAR             ;NEXT WORD MUST BE VAR
09BE 7B                    MOV  A,E             ;TEST TYPE
09BF B7                    ORA  A               ;SET FLAGS
09C0 F2F909                JP   CHA2            ;BRIF NON-STRING
09C3 CD3F1B                CALL SEARC           ;GET ADDR
09C6 D5                    PUSH D               ;SAVE IT
09C7 116F1F                LXI  D,TOLIT         ;POINT 'TO'
09CA D7                    RST  2               ;COMPARE
09CB C2A01A                JNZ  SNERR           ;BRIF ERROR
09CE CDD41B                CALL VAR             ;GET NEXT VARIABLE
09D1 7A                    MOV  A,D             ;GET HI NAME
09D2 F680                  ORI  128             ;SET MASK FOR ARRAY
09D4 57                    MOV  D,A             ;REPLACE
09D5 CD3F1B                CALL SEARC           ;GET ADDRESS
09D8 FF                    RST  7               ;POINT START OF ELEMENT 0,0
09D9 F5                    DB   -11
09DA D1                    POP  D               ;GET PTR TO STMT
09DB EB                    XCHG                 ;FLIP
09DC CD681A                CALL EOL             ;NEXT MUST BE E-O-L
09DF EB                    XCHG                 ;FLIP AGAIN
09E0 D1                    POP  D               ;GET ADDR STRING
09E1 1A                    LDAX D               ;GET COUNT
09E2 47                    MOV  B,A             ;SAVE IT
09E3 04                    INR  B               ;BUMP
09E4 C5            CHA1:   PUSH B               ;SAVE CTR
09E5 D5                    PUSH D               ;SAVE ADDR STRING
09E6 E5                    PUSH H               ;SAVE ADDR NUM
09E7 CD7012                CALL FDEC            ;CONVERT TO F.P.
09EA E1                    POP  H               ;GET ADDR
09EB F7                    RST  6               ;STORE IT
09EC FF                    RST  7               ;POINT TO NEXT
09ED F8                    DB   -8
09EE D1                    POP  D               ;RESTORE STRING
09EF C1                    POP  B               ;AND CTR
09F0 13                    INX  D               ;POINT NEXT CHAR
09F1 1A                    LDAX D               ;LOAD IT
09F2 05                    DCR  B               ;DECR CTR
09F3 C2E409                JNZ  CHA1            ;LOOP
09F6 C3F401                JMP  RUN
                   ;
                   ;
09F9 7A            CHA2:   MOV  A,D             ;GET HI NAME
09FA F680                  ORI  128             ;MAKE ARRAY NAME
09FC 57                    MOV  D,A             ;SAVE
09FD CD3F1B                CALL SEARC           ;GET ADDR
```

```
ADDR OBJECT    LINE TAG     OPCD OPRNDS         COMMENTS

0A00 FF        1622         RST  7              ;POINT ELEMENT 0,0
0A01 F5        1623         DB   -11
0A02 E3        1624         XTHL                ;SAVE ON STACK
0A03 116F1F    1625         LXI  D,TOLIT        ;POINT 'TO'
0A06 D7        1626         RST  2              ;COMPARE
0A07 C2A01A    1627         JNZ  SNERR          ;BRIF ERROR
0A0A CDD41B    1628         CALL VAR            ;GET NAME
0A0D 7B        1629         MOV  A,E            ;GET TYPE
0A0E B7        1630         ORA  A              ;SET FLAGS
0A0F F2A01A    1631         JP   SNERR          ;BRIF NOT STRING
0A12 CD681A    1632         CALL EOL            ;BRIF NOT E-O-L
0A15 CD3F1B    1633         CALL SEARC          ;GET ADDR
0A18 E1        1634         POP  H              ;GET ADDR VAR
0A19 D5        1635         PUSH D              ;SAVE D,E
0A1A 115820    1636         LXI  D,STRIN        ;POINT STRING BUFFER
0A1D D5        1637         PUSH D              ;SAVE IT
0A1E EF        1638         RST  5              ;LOAD IT
0A1F FF        1639         RST  7              ;POINT NEXT
0A20 F8        1640         DB   -8
0A21 E5        1641         PUSH H              ;SAVE H,L
0A22 CD3B1C    1642         CALL FBIN           ;CONVERT
0A25 E1        1643         POP  H              ;RESTORE
0A26 D1        1644         POP  D              ;DITTO
0A27 47        1645         MOV  B,A            ;SAVE COUNT
0A28 04        1646         INR  B              ;BUMP IT
0A29 12        1647 CHA3:   STAX D              ;PUT TO STRING
0A2A 13        1648         INX  D              ;POINT NEXT STR LOC.
0A2B C5        1649         PUSH B              ;SAVE CTRS
0A2C D5        1650         PUSH D              ;AND D ADDR
0A2D EF        1651         RST  5              ;LOAD NEXT
0A2E FF        1652         RST  7              ;POINT NEXT
0A2F F8        1653         DB   -8
0A30 E5        1654         PUSH H              ;AND H ADDR
0A31 CD3B1C    1655         CALL FBIN           ;CONVERT
0A34 E1        1656         POP  H              ;RESTORE H,L
0A35 D1        1657         POP  D              ;AND D,E
0A36 C1        1658         POP  B              ;AND CTRS
0A37 05        1659         DCR  B              ;DECR CTR
0A38 C2290A    1660         JNZ  CHA3           ;LOOP
0A3B E1        1661         POP  H              ;GET ADDR OF VAR (STRING)
0A3C CD4E06    1662         CALL LET2A          ;GO STORE IT
0A3F C3F401    1663         JMP  RUN            ;CONTINUE
```

```
ADDR OBJECT   TAG     OPCD OPRNDS          COMMENTS

0A42          DIM:    EQU  $
              ;
              ; STMT: DIM VAR(A,B),...
              ;
              ;
0A42 CDD41B           CALL VAR             ;GO GET VAR NAME
0A45 F2A01A           JP   SNERR           ;BRIF NO (
0A48 CD3F1B           CALL SEARC           ;GO LOCATE THE VAR
0A4B E3               XTHL                 ;PUT ADDR IN STACK, GET PTR TO (
0A4C F5               PUSH PSW             ;SAVE STATUS
0A4D 3EFF             MVI  A,255           ;TURN ON SW
0A4F 326821           STA  DIMSW           ;SET IT
0A52 CD0714           CALL EXPR            ;GO EVALUATE
0A55 F1               POP  PSW             ;GET STATUS
0A56 E3               XTHL                 ;SWAP PTRS
0A57 D5               PUSH D               ;SAVE ROW NUMBER
0A58 C5               PUSH B               ;SAVE COL NUMBER
0A59 03               INX  B               ;INCREMENT COLUMNS
0A5A 13               INX  D               ;AND ROWS
0A5B E5               PUSH H               ;SAVE H,L
0A5C F5               PUSH PSW             ;RESAVE STATUS
0A5D 210000           LXI  H,0             ;GET A ZERO
0A60 19       DIM1:   DAD  D               ;TIMES ONE
0A61 0B               DCX  B               ;DCR COLS
0A62 78               MOV  A,B             ;GET HI
0A63 B1               ORA  C               ;PLUS LO
0A64 C2600A           JNZ  DIM1            ;LOOP
0A67 F1               POP  PSW             ;GET STATUS
0A68 D1               POP  D               ;GET ADDRESS
0A69 29               DAD  H               ;TIMES TWO
0A6A 29               DAD  H               ;TIMES FOUR
0A6B 010800           LXI  B,8             ;PLUS 2 (NAME AND DISP)
0A6E FAAE0A           JM   REDIM           ;GO RE-DIMENSION
0A71 E5               PUSH H               ;SAVE PRODUCT
0A72 09               DAD  B               ;ADD IT
0A73 EB               XCHG                 ;FLIP/FLOP
0A74 2B               DCX  H               ;POINT LO NAME
0A75 2B               DCX  H               ;POINT HI DISP
0A76 7B               MOV  A,E             ;GET LO
0A77 2F               CMA                  ;COMPLEMENT
0A78 C601             ADI  1               ;PLUS ONE
0A7A 5F               MOV  E,A             ;RESTORE
0A7B 7A               MOV  A,D             ;GET HI
0A7C 2F               CMA                  ;COMPLEMENT
0A7D CE00             ACI  0               ;PLUS CARRY
0A7F 77               MOV  M,A             ;STORE IT
0A80 2B               DCX  H               ;POINT NEXT
0A81 73               MOV  M,E             ;STORE LO
0A82 EB               XCHG                 ;SAVE IN D,E
0A83 E1               POP  H               ;GET PRODUCT
0A84 44               MOV  B,H             ;COPY H,L
```

```
ADDR OBJECT     LINE TAG     OPCD OPRNDS        COMMENTS

0A85 4D                      MOV  C,L           ;TO B,C
0A86 EB                      XCHG               ;GET LOCAT
0A87 D1                      POP  D             ;GET COLUMNS
0A88 2B                      DCX  H             ;POINT NEXT
0A89 72                      MOV  M,D           ;MOVE LO COL
0A8A 2B                      DCX  H             ;POINT NEXT
0A8B 73                      MOV  M,E           ;MOVE HI COL
0A8C D1                      POP  D             ;GET ROWS
0A8D 2B                      DCX  H             ;POINT NEXT
0A8E 72                      MOV  M,D           ;MOVE HI ROW
0A8F 2B                      DCX  H             ;POINT NEXT
0A90 73                      MOV  M,E           ;MOVE LO ROW
0A91 2B                      DCX  H             ;POINT NEXT
0A92 3600            DIM2:   MVI  M,0           ;CLEAR ONE BYTE
0A94 2B                      DCX  H             ;POINT NEXT
0A95 0B                      DCX  B             ;DECR CTR
0A96 78                      MOV  A,B           ;GET HI
0A97 B1                      ORA  C             ;PLUS LO
0A98 C2920A                  JNZ  DIM2          ;LOOP
0A9B 3600                    MVI  M,0           ;MARK END
0A9D E1              DIM3:   POP  H             ;GET PTR TO STMT
0A9E 7E                      MOV  A,M           ;LOAD CHAR
0A9F FE2C                    CPI  ','           ;TEST IF COMMA
0AA1 C2A80A                  JNZ  DIM4          ;BRIF NOT
0AA4 23                      INX  H             ;SKIP IT
0AA5 C3420A                  JMP  DIM           ;CONTINUE
0AA8 CD681A          DIM4:   CALL EOL           ;TEST END OF LINE
0AAB C3F401                  JMP  RUN           ;CONTINUE WITH PROGRAM
0AAE 09              REDIM:  DAD  B             ;COMPUTE LEN TO NEXT
0AAF 1B                      DCX  D             ;POINT LO NAME
0AB0 1B                      DCX  D             ;POINT HI DISP
0AB1 1A                      LDAX D             ;GET IT
0AB2 47                      MOV  B,A           ;SAVE
0AB3 1B                      DCX  D             ;POINT LO DISP
0AB4 1A                      LDAX D             ;GET IT
0AB5 4F                      MOV  C,A           ;SAVE
0AB6 09                      DAD  B             ;COMPUTE DIFF BETWEEN PRIOR DIM
0AB7 7C                      MOV  A,H           ;GET HI DIFF
0AB8 B7                      ORA  A             ;TEST IT
0AB9 FAC40A                  JM   REDM1         ;BRIF PREV > NEW
0ABC C2A01A                  JNZ  SNERR         ;BRIF PREV < NEW
0ABF 7D                      MOV  A,L           ;GET LO DIFF
0AC0 B7                      ORA  A             ;TEST IT
0AC1 C2A01A                  JNZ  SNERR         ;BRIF PREV < NEW
0AC4 EB              REDM1:  XCHG               ;PUT ADDR IN H,L
0AC5 2B                      DCX  H             ;POINT HI COL
0AC6 D1                      POP  D             ;GET COL
0AC7 72                      MOV  M,D           ;MOVE HI
0AC8 2B                      DCX  H             ;POINT LO COL
0AC9 73                      MOV  M,E           ;MOVE LO
0ACA D1                      POP  D             ;GET ROW
```



```
ADDR OBJECT     LINE TAG     OPCD OPRNDS        COMMENTS

0ACB 2B                      DCX  H             ;POINT HI ROW
0ACC 72                      MOV  M,D           ;MOVE HI
0ACD 2B                      DCX  H             ;POINT LO ROW
0ACE 73                      MOV  M,E           ;MOVE LO
0ACF C39D0A                  JMP  DIM3          ;CONTINUE
```

```
ADDR OBJECT    LINE TAG      OPCD OPRNDS              COMMENTS

0AD2           1773 SUBSC:  EQU  $
               1774 ;
               1775 ;
               1776 ;  COMPUTES SUBSCR ADDR
               1777 ;  INPUT:  B HAS ROW NUMBER (1ST SUB)
               1778 ;          D HAS COL NUMBER (2ND SUB)
               1779 ;          H HAS ADDR NAME
               1780 ;
0AD2 D5        1781         PUSH D                    ;SAVE COL
0AD3 FF        1782         RST  7                    ;ADJUST H,L
0AD4 FC        1783         DB   -4                   ;BY FOUR
0AD5 56        1784         MOV  D,M                  ;GET HI
0AD6 2B        1785         DCX  H                    ;POINT LO
0AD7 5E        1786         MOV  E,M                  ;GET LO
0AD8 7A        1787         MOV  A,D                  ;GET HI
0AD9 B8        1788         CMP  B                    ;COMPARE
0ADA DAA01A    1789         JC   SNERR                ;BRIF EXCESS
0ADD C2E50A    1790         JNZ  SUB1                 ;BRIF NOT EQUAL
0AE0 7B        1791         MOV  A,E                  ;GET LO
0AE1 B9        1792         CMP  C                    ;COMPARE
0AE2 DAA01A    1793         JC   SNERR                ;BRIF EXCESS
0AE5 2B        1794 SUB1:   DCX  H                    ;POINT HI COLS
0AE6 56        1795         MOV  D,M                  ;LOAD IT
0AE7 2B        1796         DCX  H                    ;POINT LO COLS
0AE8 5E        1797         MOV  E,M                  ;LOAD IT
0AE9 E3        1798         XTHL                      ;SAVE ADDRESS
0AEA E5        1799         PUSH H                    ;SAVE SUB COL
0AEB D5        1800         PUSH D                    ;SAVE DIM COLS
0AEC 13        1801         INX  D                    ;MAKE COLS = MAX + 1 (ACCOUNT FO
0AED 210000    1802         LXI  H,0                  ;GET A ZERO
0AF0 78        1803 SUB2:   MOV  A,B                  ;GET HI
0AF1 B1        1804         ORA  C                    ;PLUS LO
0AF2 CAFA0A    1805         JZ   SUB3                 ;BRIF ZERO
0AF5 19        1806         DAD  D                    ;ADD ONCE
0AF6 0B        1807         DCX  B                    ;SUB ONCE
0AF7 C3F00A    1808         JMP  SUB2                 ;LOOP
0AFA D1        1809 SUB3:   POP  D                    ;GET DIM COL
0AFB C1        1810         POP  B                    ;GET SUB COL
0AFC 7A        1811         MOV  A,D                  ;GET HI
0AFD B8        1812         CMP  B                    ;COMPARE
0AFE DAA01A    1813         JC   SNERR                ;BRIF GT
0B01 C2090B    1814         JNZ  SUB4                 ;BRIF NOT ZERO
0B04 7B        1815         MOV  A,E                  ;GET LO
0B05 B9        1816         CMP  C                    ;COMPARE
0B06 DAA01A    1817         JC   SNERR                ;BRIF GT
0B09 09        1818 SUB4:   DAD  B                    ;ADD TO PROD
0B0A 29        1819         DAD  H                    ;TIMES TWO
0B0B 29        1820         DAD  H                    ;TIMES FOUR
0B0C 7D        1821         MOV  A,L                  ;GET LOW
0B0D 2F        1822         CMA                       ;COMPLEMENT
0B0E C601      1823         ADI  1                    ;PLUS ONE
```



```
ADDR OBJECT    LINE TAG      OPCD OPRNDS              COMMENTS

0B10 5F        1824         MOV  E,A                  ;SAVE IT
0B11 7C        1825         MOV  A,H                  ;GET HI
0B12 2F        1826         CMA                       ;COMPLEMENT
0B13 CE00      1827         ACI  0                    ;PLUS CARRY
0B15 57        1828         MOV  D,A                  ;SAVE
0B16 E1        1829         POP  H                    ;GET ADDR (0,0)
0B17 19        1830         DAD  D                    ;COMPUTE (I,J) RIGHT SIDE
0B18 FF        1831         RST  7                    ;ADJUST H,L
0B19 FC        1832         DB   -4
0B1A C9        1833         RET                       ;RETURN
```

```
ADDR OBJECT   LINE TAG    OPCD OPRNDS         COMMENTS

0B1B               FIN:   EQU  $
                        ;
                        ;  FLOATING POINT INPUT CONVERSION ROUTINE
                        ;
                        ;  THIS SUBROUTINE CONVERTS AN ASCII STRING OF CHARACTERS TO T
                        ;  POINT ACCUMULATOR.   THE INPUT FIELD MAY CONTAIN ANY VALID N
                        ;  INCLUDING SCIENTIFIC (NNN.NNNNE+NN)
                        ;  THE INPUT STRING IS TERMINATED WITH ANY NON-NUMERIC CHARACT
                        ;
                        ;
0B1B EB                   XCHG                ;PUT ADDR TO D,E
0B1C 214820               LXI  H,FACC         ;POINT FLOAT ACC
0B1F 0605                 MVI  B,5            ;BYTE COUNT
0B21 CD331C               CALL ZEROM          ;GO ZERO IT
0B24 CD8E0B               CALL FIN8           ;GET INTEGER PORTION
0B27 0E00                 MVI  C,0            ;INIT DIGIT COUNT
0B29 FE2E                 CPI  '.'            ;TEST IF DEC-POINT
0B2B C2310B               JNZ  FIN2           ;BRIF NOT
0B2E CD9E0B               CALL FIN9           ;GET FRACTION
0B31 3A4C20        FIN2:  LDA  FTEMP          ;GET EXCESS COUNT
0B34 81                   ADD  C              ;PLUS FRACTION COUNT
0B35 4F                   MOV  C,A            ;SAVE IT
0B36 FE45                 CPI  'E'            ;TEST IF EXPONENT
0B38 C2670B               JNZ  FIN4           ;BRIF NOT
0B3B D5                   PUSH D              ;SAVE PTR
0B3C 214C20               LXI  H,FTEMP        ;POINT SAVE AREA
0B3F 71                   MOV  M,C            ;SAVE DIGIT COUNT
0B40 23                   INX  H              ;POINT NEXT
0B41 F7                   RST  6              ;SAVE ACC
0B42 214820               LXI  H,FACC         ;POINT FACC
0B45 0604                 MVI  B,4            ;BYTE COUNT
0B47 CD331C               CALL ZEROM          ;GO ZERO IT
0B4A D1                   POP  D              ;RESTORE PTR
0B4B 13                   INX  D              ;SKIP 'E'
0B4C CD8E0B               CALL FIN8           ;GET NUMERIC EXP
0B4F 3A4B20               LDA  FACC+3         ;GET INTEGER
0B52 4F                   MOV  C,A            ;SAVE IT
0B53 3A4820               LDA  FACC           ;GET SIGN
0B56 B7                   ORA  A              ;TEST IT
0B57 F25D0B               JP   FIN3           ;BRIF POSITIVE
0B5A AF                   XRA  A              ;CLEAR ACC
0B5B 91                   SUB  C              ;REVERSE SIGN
0B5C 4F                   MOV  C,A            ;RESTORE
0B5D 214C20        FIN3:  LXI  H,FTEMP        ;POINT SAVED NUM
0B60 79                   MOV  A,C            ;GET DEC EXP
0B61 86                   ADD  M              ;PLUS PREV.
0B62 4F                   MOV  C,A            ;SAVE IT
0B63 23                   INX  H              ;POINT NUMBER
0B64 D5                   PUSH D              ;SAVE PTR
0B65 EF                   RST  5              ;LOAD IT
0B66 D1                   POP  D              ;RESTORE PTR
```

```
ADDR OBJECT   TAG    OPCD OPRNDS        COMMENTS

0B67 214820   FIN4:  LXI  H,FACC        ;POINT SIGN
0B6A 7E              MOV  A,M           ;LOAD IT
0B6B F618            ORI  24            ;SET UP FOR FLOAT
0B6D 77              MOV  M,A           ;RESTORE
0B6E CD1D0E          CALL FNORM         ;GO NORMALIZE
0B71 79              MOV  A,C           ;GET DEC. EXP.
0B72 EB              XCHG               ;RESTORE H,L PTR
0B73 B7       FIN5:  ORA  A             ;TEST DEC EXP
0B74 C8              RZ                 ;RETURN IF ZERO
0B75 E5              PUSH H             ;SAVE H,L
0B76 21461E          LXI  H,TEN         ;POINT CONSTANT: 10
0B79 FA860B          JM   FIN7          ;BRIF DIVIDE NEEDED
0B7C 3D              DCR  A             ;DECR COUNT
0B7D F5              PUSH PSW           ;SAVE COUNT
0B7E CD6C0E          CALL FMUL          ;GO MULTIPLY BY 10
0B81 F1       FIN6:  POP  PSW           ;RESTORE COUNT
0B82 E1              POP  H             ;RESTORE H,L
0B83 C3730B          JMP  FIN5          ;CONTINUE
0B86 3C       FIN7:  INR  A             ;INCR COUNT
0B87 F5              PUSH PSW           ;SAVE COUNT
0B88 CDE30E          CALL FDIV          ;GO DIVIDE BY 10
0B8B C3810B          JMP  FIN6          ;LOOP
0B8E 1A       FIN8:  LDAX D             ;GET CHAR
0B8F FE2B            CPI  '+'           ;TEST IF PLUS
0B91 CA9E0B          JZ   FIN9          ;BRIF IS
0B94 FE2D            CPI  '-'           ;ELSE, TEST IF NEG
0B96 C2A00B          JNZ  FINA          ;BRIF NOT
0B99 3E80            MVI  A,128         ;TEST SIGN BIT
0B9B 324820          STA  FACC          ;STORE IT
0B9E 13       FIN9:  INX  D             ;POINT NEXT
0B9F 1A              LDAX D             ;GET CHAR
0BA0 FE30     FINA:  CPI  '0'           ;TEST IF LESS ZERO
0BA2 D8              RC                 ;RETURN IF IS
0BA3 FE3A            CPI  '9'+1         ;TEST IF GT NINE
0BA5 D0              RNC                ;RETURN IF IS
0BA6 0D              DCR  C             ;COUNT THE DIGIT
0BA7 3A4920          LDA  FACC+1        ;GET MSB
0BAA E6F0            ANI  240           ;LEFT HALF
0BAC CAB60B          JZ   FINB          ;BRIF ZERO
0BAF 214C20          LXI  H,FTEMP       ;POINT EXCESS COUNT
0BB2 34              INR  M             ;ADD ONE
0BB3 C39E0B          JMP  FIN9          ;LOOP
0BB6 CDCE0B   FINB:  CALL FINC          ;GO MULT BY 10
0BB9 1A              LDAX D             ;GET THE DIGIT
0BBA E60F            ANI  15            ;MASK OFF ZONE
0BBC 214B20          LXI  H,FACC+3      ;POINT ACC
0BBF 86              ADD  M             ;ADD
0BC0 77              MOV  M,A           ;STORE
0BC1 2B              DCX  H             ;POINT NEXT
0BC2 7E              MOV  A,M           ;LOAD
0BC3 CE00            ACI  0             ;PLUS CARRY
```



```
ADDR OBJECT   TAG    OPCD OPRNDS        COMMENTS

0BC5 77              MOV  M,A           ;STORE
0BC6 2B              DCX  H             ;POINT NEXT
0BC7 7E              MOV  A,M           ;LOAD
0BC8 CE00            ACI  0             ;PLUS CARRY
0BCA 77              MOV  M,A           ;STORE
0BCB C39E0B          JMP  FIN9          ;LOOP
0BCE D5       FINC:  PUSH D             ;SAVE PTR
0BCF CDE90B          CALL FIND          ;TIMES TWO
0BD2 215120          LXI  H,FTEMP+5     ;POINT TEMP
0BD5 F7              RST  6             ;SAVE IT
0BD6 CDE90B          CALL FIND          ;TIMES THREE
0BD9 CDE90B          CALL FIND          ;TIMES FOUR
0BDC 114B20          LXI  D,FACC+3      ;POINT RESULT
0BDF 215420          LXI  H,FTEMP+8     ;POINT SAVED
0BE2 0603            MVI  B,3           ;BYTE COUNT
0BE4 CD950F          CALL FADDT         ;TIMES TEN'
0BE7 D1              POP  D             ;RESTORE PTR
0BE8 C9              RET                ;RETURN
0BE9 214B20   FIND:  LXI  H,FACC+3      ;POINT FACC
0BEC 54              MOV  D,H           ;COPY H,L
0BED 5D              MOV  E,L           ;TO D,E
0BEE 0603            MVI  B,3           ;BYTE  COUNT
0BF0 C3950F          JMP  FADDT         ;ADD & RETURN
```

```
ADDR OBJECT   LINE TAG     OPCD OPRNDS           COMMENTS

0BF3               FOUT:   EQU  $
                   ;
                   ; FLOATING POINT OUTPUT FORMAT ROUTINE
                   ;
                   ; THIS SUBROUTINE CONVERTS A NUMBER IN THE FLOATING POINT ACC
                   ; TO A FORMAT SUITABLE FOR PRINTING. THAT IS, THE NUMBER WILL
                   ; SCIENTIFIC NOTATION (+N.NNNNNE+NN) IF THE EXPONENT IS > 5 O
                   ; OTHER WISE IT WILL BE ZERO SUPPRESSED BOTH ON THE LEFT OF T
                   ; PORTION AND ON THE RIGHT OF THE FRACTION.
                   ;
0BF3 114B20                LXI  D,FACC+3         ;POINT LSB
0BF6 1A                    LDAX D                ;LOAD IT
0BF7 F607                  ORI  7                ;MASK FOR OUTPUT
0BF9 12                    STAX D                ;REPLACE
0BFA CD750F                CALL FTEST            ;GET SIGN OF NUMBER
0BFD 3620                  MVI  M,' '            ;DEFAULT SPACE
0BFF F2040C                JP   FOUT0            ;BRIF NOT MINUS
0C02 362D                  MVI  M,'-'            ;MOVE DASH
0C04 23            FOUT0:  INX  H                ;POINT NEXT
0C05 C20E0C                JNZ  FOUT2            ;BRIF NOT ZERO
0C08 3630                  MVI  M,'0'            ;MOVE THE ZERO
0C0A 23                    INX  H                ;POINT NEXT
0C0B 3620          FOUT1:  MVI  M,' '            ;MOVE SPACE FOLLOWING
0C0D C9                    RET                   ;RETURN
0C0E 3A4820        FOUT2:  LDA  FACC             ;GET SIGN & EXP
0C11 CD830F                CALL FEXP             ;EXPAND EXPONENT
0C14 C2190C                JNZ  FOUTV            ;BRIF NOT ZERO
0C17 3E80                  MVI  A,128            ;SET NEG
0C19 E680          FOUTV:  ANI  128              ;ISOLATE
0C1B 326A21                STA  DEXP             ;SAVE SIGN
0C1E E5                    PUSH H                ;SAVE H,L
0C1F 3A4820        FOUT3:  LDA  FACC             ;GET SIGN & EXP
0C22 CD830F                CALL FEXP             ;EXPAND EXP
0C25 FE01                  CPI  1                ;TEST RANGE
0C27 F2400C                JP   FOUT6            ;BRIF IN RANGE
0C2A 216A21        FOUT4:  LXI  H,DEXP           ;POINT DEC. EXP
0C2D 34                    INR  M                ;INCR IT
0C2E 21461E                LXI  H,TEN            ;POINT CONST: 10
0C31 F23A0C                JP   FOUT5            ;BRIF POS.
0C34 CD6C0E                CALL FMUL             ;MULTIPLY
0C37 C31F0C                JMP  FOUT3            ;LOOP
0C3A CDE30E        FOUT5:  CALL FDIV             ;DIVIDE
0C3D C31F0C                JMP  FOUT3            ;LOOP
0C40 FE05          FOUT6:  CPI  5                ;TEST HIGH RANGE
0C42 F22A0C                JP   FOUT4            ;BRIF 5 OR GREATER
0C45 214C20                LXI  H,FTEMP          ;POINT SAVE AREA
0C48 F7                    RST  6                ;STORE IT
0C49 3A4820                LDA  FACC             ;GET EXPONENT
0C4C CD830F                CALL FEXP             ;EXPAND
0C4F 0E06                  MVI  C,6              ;DIGIT COUNT
0C51 CDA40C                CALL FOUTB            ;SHIFT LEFT
```

```
ADDR OBJECT   TAG     OPCD OPRNDS      COMMENTS

0C54 FE0A             CPI  10          ;TEST IF DECIMAL DIGIT
0C56 FA600C           JM   FOUTU       ;BRIF LT
0C59 214C20           LXI  H,FTEMP     ;POINT SAVE AREA
0C5C EF               RST  5           ;LOAD IT
0C5D C32A0C           JMP  FOUT4       ;ONCE MORE
0C60 CD920C   FOUTU:  CALL FOUT9       ;PUT DIGIT
0C63 AF       FOUT7:  XRA  A           ;CLEAR STATUS
0C64 324820           STA  FACC        ;AND OVERFLOW
0C67 CD880C           CALL FOUT8       ;TIMES 2
0C6A 214C20           LXI  H,FTEMP     ;POINT TEMP
0C6D F7               RST  6           ;SAVE
0C6E CD880C           CALL FOUT8       ;TIMES 4
0C71 CD880C           CALL FOUT8       ;TIMES 8
0C74 114B20           LXI  D,FACC+3    ;POINT *8
0C77 214F20           LXI  H,FTEMP+3   ;POINT *2
0C7A 0604             MVI  B,4         ;BYTE COUNT
0C7C CD950F           CALL FADDT       ;GO TIMES 10
0C7F CD920C           CALL FOUT9       ;PUT DIGIT
0C82 C2630C           JNZ  FOUT7       ;LOOP
0C85 C3BB0C           JMP  FOUTH       ;GO AROUND
0C88 214B20   FOUT8:  LXI  H,FACC+3    ;POINT NUMBER
0C8B 54               MOV  D,H         ;COPY H,L
0C8C 5D               MOV  E,L         ;TO D,E
0C8D 0604             MVI  B,4         ;BYTE COUNT
0C8F C3950F           JMP  FADDT       ;GO ADD
0C92 F630     FOUT9:  ORI  48          ;DEC. ZONE
0C94 E1               POP  H           ;GET RETURN ADDR
0C95 E3               XTHL             ;EXCH WITH TOP (PTR)
0C96 77               MOV  M,A         ;PUT DIGIT
0C97 23               INX  H           ;POINT NEXT
0C98 79               MOV  A,C         ;GET COUNT
0C99 FE06             CPI  6           ;TEST IF 1ST
0C9B C2A10C           JNZ  FOUTA       ;BRIF NOT
0C9E 362E             MVI  M,'.'       ;MOVE DEC. PT.
0CA0 23               INX  H           ;POINT NEXT
0CA1 E3       FOUTA:  XTHL             ;EXCH WITH RTN
0CA2 0D               DCR  C           ;DECR COUNT
0CA3 E9               PCHL             ;RETURN
0CA4 5F       FOUTB:  MOV  E,A         ;SAVE BIT COUNT
0CA5 AF               XRA  A           ;CLEAR ACC FLAGS
0CA6 324820           STA  FACC        ;AND OVERFLOW
0CA9 214B20   FOUTC:  LXI  H,FACC+3    ;POINT LSB
0CAC 0604             MVI  B,4         ;BYTE COUNT
0CAE 7E       FOUTD:  MOV  A,M         ;GET A BYTE
0CAF 17               RAL              ;SHIFT LEFT
0CB0 77               MOV  M,A         ;STORE
0CB1 2B               DCX  H           ;POINT NEXT
0CB2 05               DCR  B           ;DECR CTR
0CB3 C2AE0C           JNZ  FOUTD       ;LOOP
0CB6 1D               DCR  E           ;DECR BIT CTR
0CB7 C2A90C           JNZ  FOUTC       ;LOOP
```

```
ADDR OBJECT   LINE TAG    OPCD OPRNDS        COMMENTS

0CBA C9                   RET                ;RETURN
0CBB E1            FOUTH: POP  H             ;GET PTR
0CBC 3645                 MVI  M,'E'         ;EXPONENT
0CBE 23                   INX  H             ;POINT NEXT
0CBF 3A6A21               LDA  DEXP          ;GET EXPONENT
0CC2 362B                 MVI  M,'+'         ;DEFAULT
0CC4 57                   MOV  D,A           ;SAVE NUMBER
0CC5 B7                   ORA  A             ;TEST IT
0CC6 F2D20C               JP   FOUTI         ;BRIF POS
0CC9 362D                 MVI  M,'-'         ;ELSE, DASH
0CCB E67F                 ANI  127           ;STRIP DUMB SIGN
0CCD 2F                   CMA                ;COMPLEMENT
0CCE 3C                   INR  A             ;PLUS ONE (TWOS COMP)
0CCF 57                   MOV  D,A           ;SAVE IT
0CD0 2F                   CMA                ;RE-COMPLEMENT
0CD1 3C                   INR  A             ;PLUS ONE
0CD2 23            FOUTI: INX  H             ;POINT NEXT
0CD3 E5                   PUSH H             ;SAVE PTR
0CD4 1EFF                 MVI  E,-1          ;INIT CTR (TENS)
0CD6 1C            FOUTJ: INR  E             ;ADD ONE
0CD7 D60A                 SUI  10            ;LESS 10
0CD9 F2D60C               JP   FOUTJ         ;LOOP
0CDC C60A                 ADI  10            ;CORRECT UNITS
0CDE 47                   MOV  B,A           ;SAVE UNITS
0CDF 7B                   MOV  A,E           ;GET TENS
0CE0 CD920C               CALL FOUT9         ;OUTPUT
0CE3 78                   MOV  A,B           ;GET UNITS
0CE4 CD920C               CALL FOUT9         ;OUTPUT
0CE7 E1                   POP  H             ;GET PTR
0CE8 3620                 MVI  M,' '         ;SPACE AFTER
0CEA 7A                   MOV  A,D           ;GET DEC EXPON
0CEB B7                   ORA  A             ;SET FLAGS
0CEC F2F50C               JP   FOUTK         ;BRIF POS.
0CEF FEFE                 CPI  -2            ;TEST FOR MIN
0CF1 D8                   RC                 ;RETURN IF LESS THAN -2
0CF2 C3F80C               JMP  FOUTL         ;GO AROUND
0CF5 FE06          FOUTK: CPI  6             ;TEST IF TOO BIG
0CF7 D0                   RNC                ;RETURN IF 6 OR GREATER
0CF8 4F            FOUTL: MOV  C,A           ;SAVE EXPONENT
0CF9 0605                 MVI  B,5           ;CTR
0CFB 3620          FOUTM: MVI  M,' '         ;SPACE OUT EXPONENT
0CFD 2B                   DCX  H             ;POINT PRIOR
0CFE 05                   DCR  B             ;DECR CTR
0CFF C2FB0C               JNZ  FOUTM         ;LOOP
0D02 EB                   XCHG               ;FLIP/FLOP
0D03 7B                   MOV  A,E           ;GET LOW BYTE
0D04 D605                 SUI  5             ;POINT TO DOT
0D06 6F                   MOV  L,A           ;PUT DOWN
0D07 7A                   MOV  A,D           ;GET HIGH
0D08 DE00                 SBI  0             ;IN CASE OF BORROW
0D0A 67                   MOV  H,A           ;PUT DOWN
```

```
ADDR OBJECT    LINE TAG     OPCD OPRNDS           COMMENTS

0D0B 79        2114         MOV  A,C              ;GET EXPONENT
0D0C B7        2115         ORA  A                ;TEST SIGN
0D0D CA1E0D    2116         JZ   FOUTO            ;BRIF ZERO
0D10 FA330D    2117         JM   FOUTR            ;BRIF NEGATIVE
0D13 46        2118 FOUTN:  MOV  B,M              ;GET HIGH BYTE
0D14 23        2119         INX  H                ;POINT NEXT
0D15 7E        2120         MOV  A,M              ;GET LOW BYTE
0D16 70        2121         MOV  M,B              ;SHIFT DOT TO RIGHT
0D17 2B        2122         DCX  H                ;POINT BACK
0D18 77        2123         MOV  M,A              ;MOVE THE DIGIT LEFT
0D19 23        2124         INX  H                ;POINT NEXT
0D1A 0D        2125         DCR  C                ;DECR CTR
0D1B C2130D    2126         JNZ  FOUTN            ;LOOP
0D1E EB        2127 FOUTO:  XCHG                  ;POINT END
0D1F 7E        2128 FOUTP:  MOV  A,M              ;GET A DIGIT/DOT
0D20 FE30      2129         CPI  '0'              ;TEST FOR TRAILING ZERO
0D22 C22B0D    2130         JNZ  FOUTQ            ;BRIF NOT
0D25 3620      2131         MVI  M,' '            ;SPACE FILL
0D27 2B        2132         DCX  H                ;POINT PRIOR
0D28 C31F0D    2133         JMP  FOUTP            ;LOOP
0D2B FE2E      2134 FOUTQ:  CPI  '.'              ;TEST FOR TRAILING DOT
0D2D 23        2135         INX  H                ;JUST IN CASE NOT
0D2E C0        2136         RNZ                   ;RETURN IF NOT
0D2F 2B        2137         DCX  H                ;RESET PTR
0D30 3620      2138         MVI  M,' '            ;SPACE IT OUT
0D32 C9        2139         RET                   ;RETURN
0D33 FEFF      2140 FOUTR:  CPI  255              ;TEST IF -1
0D35 C2410D    2141         JNZ  FOUTS            ;ELSE -2
0D38 2B        2142         DCX  H                ;POINT SIGNIFICANT
0D39 7E        2143         MOV  A,M              ;GET THE CHAR
0D3A 362E      2144         MVI  M,'.'            ;MOVE THE DOT
0D3C 23        2145         INX  H                ;POINT NEXT
0D3D 77        2146         MOV  M,A              ;SHIFT THE DIGIT
0D3E C31E0D    2147         JMP  FOUTO            ;GO ZERO SUPPRESS
0D41 2B        2148 FOUTS:  DCX  H                ;POINT ONE TO LEFT
0D42 7E        2149         MOV  A,M              ;PICK UP DIGIT
0D43 3630      2150         MVI  M,'0'            ;REPLACE
0D45 23        2151         INX  H                ;POINT RIGHT
0D46 77        2152         MOV  M,A              ;PUT THE DIGIT
0D47 62        2153         MOV  H,D              ;GET LOW ADDR
0D48 6B        2154         MOV  L,E              ;POINT LAST DIGIT
0D49 0606      2155         MVI  B,6              ;CTR
0D4B 2B        2156 FOUTT:  DCX  H                ;POINT PRITO
0D4C 7E        2157         MOV  A,M              ;GET A DIGIT
0D4D 23        2158         INX  H                ;POINT
0D4E 77        2159         MOV  M,A              ;PUT IT ONE TO RIGHT
0D4F 2B        2160         DCX  H                ;POINT
0D50 05        2161         DCR  B                ;DECR CTR
0D51 C24B0D    2162         JNZ  FOUTT            ;LOOP
0D54 362E      2163         MVI  M,'.'            ;MOVE THE DOT
0D56 C31E0D    2164         JMP  FOUTO            ;CONTINUE
```



```
ADDR OBJECT    LINE TAG     OPCD OPRNDS           COMMENTS
```

```
ADDR OBJECT   TAG     OPCD OPRNDS       COMMENTS

0CBA C9               RET               ;RETURN
0CBB E1       FOUTH:  POP  H            ;GET PTR
0CBC 3645             MVI  M,'E'        ;EXPONENT
0CBE 23               INX  H            ;POINT NEXT
0CBF 3A6A21           LDA  DEXP         ;GET EXPONENT
0CC2 362B             MVI  M,'+'        ;DEFAULT
0CC4 57               MOV  D,A          ;SAVE NUMBER
0CC5 B7               ORA  A            ;TEST IT
0CC6 F2D20C           JP   FOUTI        ;BRIF POS
0CC9 362D             MVI  M,'-'        ;ELSE, DASH
0CCB E67F             ANI  127          ;STRIP DUMB SIGN
0CCD 2F               CMA               ;COMPLEMENT
0CCE 3C               INR  A            ;PLUS ONE (TWOS COMP)
0CCF 57               MOV  D,A          ;SAVE IT
0CD0 2F               CMA               ;RE-COMPLEMENT
0CD1 3C               INR  A            ;PLUS ONE
0CD2 23       FOUTI:  INX  H            ;POINT NEXT
0CD3 E5               PUSH H            ;SAVE PTR
0CD4 1EFF             MVI  E,-1         ;INIT CTR (TENS)
0CD6 1C       FOUTJ:  INR  E            ;ADD ONE
0CD7 D60A             SUI  10           ;LESS 10
0CD9 F2D60C           JP   FOUTJ        ;LOOP
0CDC C60A             ADI  10           ;CORRECT UNITS
0CDE 47               MOV  B,A          ;SAVE UNITS
0CDF 7B               MOV  A,E          ;GET TENS
0CE0 CD920C           CALL FOUT9        ;OUTPUT
0CE3 78               MOV  A,B          ;GET UNITS
0CE4 CD920C           CALL FOUT9        ;OUTPUT
0CE7 E1               POP  H            ;GET PTR
0CE8 3620             MVI  M,' '        ;SPACE AFTER
0CEA 7A               MOV  A,D          ;GET DEC EXPON
0CEB B7               ORA  A            ;SET FLAGS
0CEC F2F50C           JP   FOUTK        ;BRIF POS.
0CEF FEFE             CPI  -2           ;TEST FOR MIN
0CF1 D8               RC                ;RETURN IF LESS THAN -2
0CF2 C3F80C           JMP  FOUTL        ;GO AROUND
0CF5 FE06     FOUTK:  CPI  6            ;TEST IF TOO BIG
0CF7 D0               RNC               ;RETURN IF 6 OR GREATER
0CF8 4F       FOUTL:  MOV  C,A          ;SAVE EXPONENT
0CF9 0605             MVI  B,5          ;CTR
0CFB 3620     FOUTM:  MVI  M,' '        ;SPACE OUT EXPONENT
0CFD 2B               DCX  H            ;POINT PRIOR
0CFE 05               DCR  B            ;DECR CTR
0CFF C2FB0C           JNZ  FOUTM        ;LOOP
0D02 EB               XCHG              ;FLIP/FLOP
0D03 7B               MOV  A,E          ;GET LOW BYTE
0D04 D605             SUI  5            ;POINT TO DOT
0D06 6F               MOV  L,A          ;PUT DOWN
0D07 7A               MOV  A,D          ;GET HIGH
0D08 DE00             SBI  0            ;IN CASE OF BORROW
0D0A 67               MOV  H,A          ;PUT DOWN
```

```
ADDR OBJECT   LINE TAG     OPCD OPRNDS       COMMENTS

0D0B 79       2114         MOV  A,C          ;GET EXPONENT
0D0C B7       2115         ORA  A            ;TEST SIGN
0D0D CA1E0D   2116         JZ   FOUTO        ;BRIF ZERO
0D10 FA330D   2117         JM   FOUTR        ;BRIF NEGATIVE
0D13 46       2118 FOUTN:  MOV  B,M          ;GET HIGH BYTE
0D14 23       2119         INX  H            ;POINT NEXT
0D15 7E       2120         MOV  A,M          ;GET LOW BYTE
0D16 70       2121         MOV  M,B          ;SHIFT DOT TO RIGHT
0D17 2B       2122         DCX  H            ;POINT BACK
0D18 77       2123         MOV  M,A          ;MOVE THE DIGIT LEFT
0D19 23       2124         INX  H            ;POINT NEXT
0D1A 0D       2125         DCR  C            ;DECR CTR
0D1B C2130D   2126         JNZ  FOUTN        ;LOOP
0D1E EB       2127 FOUTO:  XCHG              ;POINT END
0D1F 7E       2128 FOUTP:  MOV  A,M          ;GET A DIGIT/DOT
0D20 FE30     2129         CPI  '0'          ;TEST FOR TRAILING ZERO
0D22 C22B0D   2130         JNZ  FOUTQ        ;BRIF NOT
0D25 3620     2131         MVI  M,' '        ;SPACE FILL
0D27 2B       2132         DCX  H            ;POINT PRIOR
0D28 C31F0D   2133         JMP  FOUTP        ;LOOP
0D2B FE2E     2134 FOUTQ:  CPI  '.'          ;TEST FOR TRAILING DOT
0D2D 23       2135         INX  H            ;JUST IN CASE NOT
0D2E C0       2136         RNZ               ;RETURN IF NOT
0D2F 2B       2137         DCX  H            ;RESET PTR
0D30 3620     2138         MVI  M,' '        ;SPACE IT OUT
0D32 C9       2139         RET               ;RETURN
0D33 FEFF     2140 FOUTR:  CPI  255          ;TEST IF -1
0D35 C2410D   2141         JNZ  FOUTS        ;ELSE -2
0D38 2B       2142         DCX  H            ;POINT SIGNIFICANT
0D39 7E       2143         MOV  A,M          ;GET THE CHAR
0D3A 362E     2144         MVI  M,'.'        ;MOVE THE DOT
0D3C 23       2145         INX  H            ;POINT NEXT
0D3D 77       2146         MOV  M,A          ;SHIFT THE DIGIT
0D3E C31E0D   2147         JMP  FOUTO        ;GO ZERO SUPPRESS
0D41 2B       2148 FOUTS:  DCX  H            ;POINT ONE TO LEFT
0D42 7E       2149         MOV  A,M          ;PICK UP DIGIT
0D43 3630     2150         MVI  M,'0'        ;REPLACE
0D45 23       2151         INX  H            ;POINT RIGHT
0D46 77       2152         MOV  M,A          ;PUT THE DIGIT
0D47 62       2153         MOV  H,D          ;GET LOW ADDR
0D48 6B       2154         MOV  L,E          ;POINT LAST DIGIT
0D49 0606     2155         MVI  B,6          ;CTR
0D4B 2B       2156 FOUTT:  DCX  H            ;POINT PRITO
0D4C 7E       2157         MOV  A,M          ;GET A DIGIT
0D4D 23       2158         INX  H            ;POINT
0D4E 77       2159         MOV  M,A          ;PUT IT ONE TO RIGHT
0D4F 2B       2160         DCX  H            ;POINT
0D50 05       2161         DCR  B            ;DECR CTR
0D51 C24B0D   2162         JNZ  FOUTT        ;LOOP
0D54 362E     2163         MVI  M,'.'        ;MOVE THE DOT
0D56 C31E0D   2164         JMP  FOUTO        ;CONTINUE
```



```
ADDR OBJECT   LINE TAG     OPCD OPRNDS       COMMENTS
```

```
ADDR OBJECT   LINE TAG    OPCD OPRNDS           COMMENTS

0D59               FADD:  EQU  $
                   ;
                   ;
                   ; FLOATING POINT ADD THE NUMBER AT (H,L) TO THE FACC
                   ;
                   ;
0D59 23                   INX  H                ;POINT FIRST DIGIT
0D5A 7E                   MOV  A,M              ;LOAD IT
0D5B B7                   ORA  A                ;TEST IT
0D5C CA750F               JZ   FTEST            ;BRIF ZERO
0D5F 2B                   DCX  H                ;POINT BACK
0D60 CD750F               CALL FTEST            ;GO TEST SIGN OF FACC
0D63 CA2800               JZ   RST5             ;JUST LOAD IF FACC = 0
0D66 114820               LXI  D,FACC           ;POINT FACC
0D69 1A                   LDAX D                ;GET EXPONENT OF FACC
0D6A CD830F               CALL FEXP             ;GO GET EXPONENT
0D6D 47                   MOV  B,A              ;SAVE EXPONENT
0D6E 7E                   MOV  A,M              ;GET EXPONENT OF ADDR
0D6F CD830F               CALL FEXP             ;GO GET EXPONENT
0D72 4F                   MOV  C,A              ;SAVE THE EXPONENT
0D73 90                   SUB  B                ;GET DIFFERENCE OF TWO EXPONENTS
0D74 CA890D               JZ   FADD4            ;BRIF THEY'RE EQUAL
0D77 F27C0D               JP   FADD3            ;BRIF DIFFERENCE IS POSITIVE
0D7A 2F                   CMA                   ;COMPLEMENT ACC
0D7B 3C                   INR  A                ;PLUS ONE (TWO'S COMPLEMENT)
0D7C FE18          FADD3: CPI  24               ;COMPARE DIFFERENCE TO MAX
0D7E DA890D               JC   FADD4            ;BRIF LESS
0D81 78                   MOV  A,B              ;GET EXPON OF ADDUEND
0D82 91                   SUB  C                ;GET TRUE DIFFERENCE AGAIN
0D83 F2750F               JP   FTEST            ;BRIF FACC > ADDER
0D86 C32800               JMP  RST5             ;ELSE, ADDER > FACC
0D89 F5            FADD4: PUSH PSW              ;SAVE DIFFERENCE
0D8A C5                   PUSH B                ;SAVE EXPONENTS
0D8B 114C20               LXI  D,FTEMP          ;GET ADDR OF TEMP ACC
0D8E 0604                 MVI  B,4              ;MOVE LENGTH
0D90 CD2D1C               CALL COPYH            ;GO COPY
0D93 C1                   POP  B                ;GET EXPONENTS
0D94 F1                   POP  PSW              ;GET DIFFERENCE
0D95 CABC0D               JZ   FADD9            ;JUST ADD IF ZERO
0D98 214D20               LXI  H,FTEMP+1        ;DEFAULT
0D9B F5                   PUSH PSW              ;SAVE DIFFERENCE
0D9C 78                   MOV  A,B              ;GET FACC EXPON
0D9D 91                   SUB  C                ;MINUS FTEMP EXPON
0D9E F2AE0D               JP   FADD6            ;BRIF TEMP MUST BE SHIFTED
0DA1 214820               LXI  H,FACC           ;POINT FLOAT ACC
0DA4 79                   MOV  A,C              ;GET EXPONENT, SIGN
0DA5 E67F                 ANI  127              ;STRIP EXP SIGN
0DA7 4F                   MOV  C,A              ;PUT BACK
0DA8 7E                   MOV  A,M              ;GET THE EXP
0DA9 E680                 ANI  128              ;STRIP OFF OLD EXPON
0DAB B1                   ORA  C                ;MOVE ADDER EXPON TO IT
```

```
ADDR OBJECT   LINE TAG     OPCD OPRNDS        COMMENTS

0DAC 77                    MOV  M,A           ;REPLACE
0DAD 23                    INX  H             ;POINT FIRST DATA BYTE
0DAE F1             FADD6: POP  PSW           ;GET DIFFER
0DAF 4F                    MOV  C,A           ;SAVE IT
0DB0 0603           FADD7: MVI  B,3           ;LOOP CTR (INNER)
0DB2 AF                    XRA  A             ;INIT CARRY TO Z
0DB3 E5                    PUSH H             ;SAVE ADDR
0DB4 CDA00F                CALL FSHFT         ;GO SHIFT
0DB7 E1                    POP  H             ;GET ADDR
0DB8 0D                    DCR  C             ;DECR CTR
0DB9 C2B00D                JNZ  FADD7         ;LOOP
0DBC                FADD9: EQU  $
0DBC 114820                LXI  D,FACC        ;POINT SIGN OF ADDUEND
0DBF 214C20                LXI  H,FTEMP       ;AND SIGN OF ADDER
0DC2 1A                    LDAX D             ;GET SIGN OF ADDUEND
0DC3 AE                    XRA  M             ;COMPARE THE TWO SIGNS
0DC4 FAEF0D                JM   FADDA         ;BRIF SIGNS DIFFER
0DC7 114B20                LXI  D,FACC+3      ;POINT LOW END
0DCA 214F20                LXI  H,FTEMP+3     ;DITTO
0DCD 0603                  MVI  B,3           ;BYTE COUNT
0DCF CD950F                CALL FADDT         ;GO ADD TWO TOGETHER
0DD2 D2750F                JNC  FTEST         ;BRIF NO CARRY
0DD5 214820                LXI  H,FACC        ;GET ADDR OF ACC
0DD8 7E                    MOV  A,M           ;LOAD THE EXPON
0DD9 E680                  ANI  128           ;ISOLATE SIGN
0DDB 47                    MOV  B,A           ;SAVE SIGN
0DDC 7E                    MOV  A,M           ;GET EXPON
0DDD CD830F                CALL FEXP          ;GO GET EXPONENT
0DE0 3C                    INR  A             ;ADD ONE
0DE1 E67F                  ANI  127           ;ISOLATE
0DE3 B0                    ORA  B             ;PUT BACK SIGN
0DE4 77                    MOV  M,A           ;PUT IT DOWN
0DE5 23                    INX  H             ;POINT DATA
0DE6 37                    STC                ;SET CY
0DE7 0603                  MVI  B,3           ;CTR
0DE9 CDA00F                CALL FSHFT         ;GO SHIFT IT
0DEC C3750F                JMP  FTEST         ;RETURN
0DEF                FADDA: EQU  $
0DEF 215020                LXI  H,FTEMP+4     ;POINT TEMP2 AREA
0DF2 0604                  MVI  B,4           ;PREPARE TO SAVE ACC
0DF4 CD241C                CALL COPYD         ;GO COPY
0DF7 114B20         FADDB: LXI  D,FACC+3      ;POINT LOW ACC
0DFA 214F20                LXI  H,FTEMP+3     ;AND LOW TEMP
0DFD 0603                  MVI  B,3           ;CTR
0DFF CD8A0F                CALL FSUBT         ;GO SUBTRACT THE TWO
0E02 D21D0E                JNC  FNORM         ;BRIF NO BORROW
0E05 114820                LXI  D,FACC        ;POINT ACC
0E08 214C20                LXI  H,FTEMP       ;POINT TEMP
0E0B 0608                  MVI  B,8           ;CTR
0E0D CD2D1C                CALL COPYH         ;GO COPY
0E10 114820                LXI  D,FACC        ;POINT
```



```
ADDR OBJECT   LINE TAG     OPCD OPRNDS        COMMENTS

0E13 214C20                LXI  H,FTEMP       ;TEMP
0E16 7E                    MOV  A,M           ;GET ORIG ACC EXPONENT
0E17 EE80                  XRI  128           ;REVERSE SIGN
0E19 12                    STAX D             ;PUT TO NEW ACC
0E1A C3F70D                JMP  FADDB         ;GO SUBTRACT AGAIN
```

```
ADDR OBJECT   LINE TAG     OPCD OPRNDS           COMMENTS

0E1D          2274 FNORM:  EQU  $
              2275 ;
              2276 ;
              2277 ;  NORMALIZE THE FLOATING ACCUMULATOR
              2278 ;  THAT IS, THE FIRST BIT MUST BE SIGNIFICANT
              2279 ;
              2280 ;
0E1D 214B20   2281         LXI  H,FACC+3         ;POINT LSB
0E20 7E       2282         MOV  A,M              ;LOAD IT
0E21 2B       2283         DCX  H                ;POINT PRIOR
0E22 B6       2284         ORA  M                ;MERGE
0E23 2B       2285         DCX  H                ;POINT PRIOR
0E24 B6       2286         ORA  M                ;MERGE
0E25 C22B0E   2287         JNZ  FNRM1            ;BRIF NOT ZERO
0E28 2B       2288         DCX  H                ;POINT SIGN&EXP
0E29 77       2289         MOV  M,A              ;SET TO ZERO
0E2A C9       2290         RET                   ;RETURN
0E2B 7E       2291 FNRM1:  MOV  A,M              ;GET MSB
0E2C B7       2292         ORA  A                ;TEST IT
0E2D FA750F   2293         JM   FTEST            ;BRIF NORMALIZED
0E30 23       2294         INX  H                ;PINT TO
0E31 23       2295         INX  H                ;LSB
0E32 0603     2296         MVI  B,3              ;GET NUMBER OF BYTES
0E34 AF       2297         XRA  A                ;CLEAR CY
0E35 7E       2298 FNRM2:  MOV  A,M              ;GET A BYTE
0E36 17       2299         RAL                   ;LEFT 1 BIT
0E37 77       2300         MOV  M,A              ;STORE THE BYTE
0E38 2B       2301         DCX  H                ;POINT NEXT
0E39 05       2302         DCR  B                ;DCR CTR
0E3A C2350E   2303         JNZ  FNRM2            ;LOOP
0E3D 7E       2304         MOV  A,M              ;GET SIGN&EXP
0E3E CD830F   2305         CALL FEXP             ;EXPAND EXP TO 8 BITS
0E41 3D       2306         DCR  A                ;MINUS ONE
0E42 FE80     2307         CPI  128              ;TEST IF UNDERFLOW
0E44 CA981A   2308         JZ   OVERR            ;BRIF IS
0E47 E67F     2309         ANI  127              ;ISOLATE LEAST 7 BITS
0E49 47       2310         MOV  B,A              ;SAVE
0E4A 7E       2311         MOV  A,M              ;GET SIGN
0E4B E680     2312         ANI  128              ;ISOLATE
0E4D B0       2313         ORA  B                ;MERGE
0E4E 77       2314         MOV  M,A              ;STORE
0E4F 23       2315         INX  H                ;POINT MSB
0E50 C32B0E   2316         JMP  FNRM1            ;LOOP
```



```
ADDR OBJECT   LINE TAG     OPCD OPRNDS           COMMENTS

0E53          2318 FSUB:   EQU  $
              2319 ;
              2320 ;
              2321 ;  FLOATING POINT SUBTRACT THE NUMBER AT (H,L) FROM THE FACC
              2322 ;
              2323 ;
0E53 23       2324         INX  H                ;POINT FIRST DATA BYTE OF SUBTRA
0E54 7E       2325         MOV  A,M              ;LOAD IT
0E55 B7       2326         ORA  A                ;TEST
0E56 CA750F   2327         JZ   FTEST            ;BRIF ZERO
0E59 2B       2328         DCX  H                ;POINT BACK
0E5A 114C20   2329         LXI  D,FTEMP          ;GET TEMPORY STORAGE AREA
0E5D 0605     2330         MVI  B,5              ;MOVE LEN
0E5F CD2D1C   2331         CALL COPYH            ;GO COPY
0E62 214C20   2332         LXI  H,FTEMP          ;POINT NEW AREA
0E65 7E       2333         MOV  A,M              ;GET EXPONENT
0E66 EE80     2334         XRI  128              ;REVERSE SIGN
0E68 77       2335         MOV  M,A              ;REPLACE
0E69 C3590D   2336         JMP  FADD             ;GO ADD THE TWO
```

```
ADDR OBJECT   LINE TAG    OPCD OPRNDS           COMMENTS

0E6C          2338 FMUL:  EQU  $
              2339 ;
              2340 ;
              2341 ;  FLOATING POINT MULTIPLY THE NUMBER AT (H,L) TO THE FACC
              2342 ;
              2343 ;
0E6C CD750F   2344        CALL FTEST            ;TEST FACC
0E6F C8       2345        RZ                    ;RETURN IF ZERO
0E70 23       2346        INX  H                ;POINT 1ST DIGIT OF MULTIPLIER
0E71 7E       2347        MOV  A,M              ;LOAD IT
0E72 2B       2348        DCX  H                ;RESTORE
0E73 B7       2349        ORA  A                ;TEST IF ZERO
0E74 CA2800   2350        JZ   RST5             ;GO LOAD TO FACC IF IT IS
0E77 114820   2351 FMUL1: LXI  D,FACC           ;POINT EXPONENT
0E7A 1A       2352        LDAX D                ;LOAD IT
0E7B CD830F   2353        CALL FEXP             ;GO GET EXPONENT
0E7E 47       2354        MOV  B,A              ;SAVE IN B
0E7F 7E       2355        MOV  A,M              ;GET EXPONENT OF MULTIPLIER
0E80 CD830F   2356        CALL FEXP             ;GO GET EXPONENT
0E83 80       2357        ADD  B                ;ADD EXPONENTS TOGETHER
0E84 CD650F   2358        CALL FOVUN            ;GO SEE IF OVERFLOW/UNDERFLOW
0E87 E67F     2359        ANI  127              ;TURN OFF SIGN
0E89 47       2360        MOV  B,A              ;SAVE
0E8A 1A       2361        LDAX D                ;GET SIGN OF FACC
0E8B AE       2362        XRA  M                ;PRODUCT SIGN IS NEG IF TWO SIGN
0E8C E680     2363        ANI  128              ;MASK
0E8E B0       2364        ORA  B                ;PUT SIGN AND EXPONENT TOGETHER
0E8F 12       2365        STAX D                ;PUT IN FACC
0E90 115520   2366        LXI  D,FTEMP+9        ;POINT TEMP STORAGE
0E93 0603     2367        MVI  B,3              ;BYTE COUNT
0E95 23       2368        INX  H                ;POINT MSD
0E96 CD2D1C   2369        CALL COPYH            ;MOVE MULTIPLIER
0E99 214C20   2370        LXI  H,FTEMP          ;POINT DIGIT 7 OF RESULT
0E9C 0606     2371        MVI  B,6              ;LOOP CTR
0E9E CD331C   2372        CALL ZEROM            ;GO ZERO EIGHT BYTES
0EA1 114920   2373        LXI  D,FACC+1         ;POINT 1ST DIGIT OF ACC
0EA4 0603     2374        MVI  B,3              ;LOOP CTR
0EA6 1A       2375 FMUL5: LDAX D                ;GET AN ACC DIGIT PAIR
0EA7 77       2376        MOV  M,A              ;PUT TO TEMP STORAGE
0EA8 AF       2377        XRA  A                ;ZERO A
0EA9 12       2378        STAX D                ;CLEAR ACC
0EAA 13       2379        INX  D                ;POINT NEXT
0EAB 23       2380        INX  H                ;DITTO
0EAC 05       2381        DCR  B                ;DECR CTR
0EAD C2A60E   2382        JNZ  FMUL5            ;LOOP
0EB0 0E18     2383        MVI  C,24             ;OUTTER LOOP CTR
0EB2 0603     2384 FMUL6: MVI  B,3              ;CTR
0EB4 215520   2385        LXI  H,FTEMP+9        ;POINT MULTIPLIER
0EB7 AF       2386        XRA  A                ;CLEAR CY
0EB8 7E       2387 FMUL7: MOV  A,M              ;GET BYTE
0EB9 1F       2388        RAR                   ;SHIFT RIGHT
```



```
ADDR OBJECT   LINE TAG    OPCD OPRNDS           COMMENTS

0EBA 77       2389        MOV  M,A              ;PUT DOWN
0EBB 23       2390        INX  H                ;POINT NEXT
0EBC 05       2391        DCR  B                ;DECR CTR
0EBD C2B80E   2392        JNZ  FMUL7            ;LOOP
0EC0 D2CE0E   2393        JNC  FMUL8            ;BRIF ZERO BIT
0EC3 114E20   2394        LXI  D,FTEMP+2        ;POINT RESULT
0EC6 215420   2395        LXI  H,FTEMP+8        ;POINT MULTIPLICAND
0EC9 0606     2396        MVI  B,6              ;SIX BYTE ADD
0ECB CD950F   2397        CALL FADDT            ;GO ADD
0ECE 0606     2398 FMUL8: MVI  B,6              ;SIZ BYTE SHIFT
0ED0 215420   2399        LXI  H,FTEMP+8        ;POINT MULTIPLICAND
0ED3 AF       2400        XRA  A                ;CLEAR CY
0ED4 7E       2401 FMUL9: MOV  A,M              ;GET BYTE
0ED5 17       2402        RAL                   ;SHIFT LEFT
0ED6 77       2403        MOV  M,A              ;PUT BACT
0ED7 2B       2404        DCX  H                ;POINT NEXT BYTE
0ED8 05       2405        DCR  B                ;DECR CTR
0ED9 C2D40E   2406        JNZ  FMUL9            ;LOOP
0EDC 0D       2407        DCR  C                ;DEC BIT COUNT
0EDD C2B20E   2408        JNZ  FMUL6            ;CONTINUE
0EE0 C3100E   2409        JMP  FNORM            ;GO NORMALIZE
```

```
ADDR OBJECT    TAG     OPCD OPRNDS          COMMENTS

0EE3           FDIV:   EQU  $
               ;
               ;
               ; FLOATING POINT DIVIDE THE NUMBER AT (H,L) INTO THE FACC
               ;
               ;
0EE3 CD750F            CALL FTEST           ;TEST IF FACC ZERO
0EE6 C8                RZ                   ;RETURN IF IT IS
0EE7 23                INX  H               ;POINT 1ST DIGIT OF DIVISOR
0EE8 7E                MOV  A,M             ;LOAD IT
0EE9 2B                DCX  H               ;POINT BACK
0EEA B7                ORA  A               ;TEST IF ZERO
0EEB CA981A            JZ   OVERR           ;DIVISION BY ZERO = ERROR
0EEE 7E                MOV  A,M             ;LOAD EXPONENT OF DIVISOR
0EEF CD830F            CALL FEXP            ;GO GET EXPON
0EF2 47                MOV  B,A             ;SAVE IT
0EF3 114820            LXI  D,FACC          ;POINT EXPONENT OF DIVIDEND
0EF6 1A                LDAX D               ;LOAD IT
0EF7 CD830F            CALL FEXP            ;GO GET EXPON
0EFA 90                SUB  B               ;SUBTRACT THE TWO EXPONENTS
0EFB 3C                INR  A               ;PLUS ONE
0EFC CD650F            CALL FOVUN           ;GO SEE IF OVERFLOW/UNDERFLOW
0EFF E67F              ANI  127             ;TRUNCATE TO 7 BITS
0F01 47                MOV  B,A             ;SAVE IT
0F02 1A                LDAX D               ;GET EXPONENT
0F03 AE                XRA  M               ;IF SIGNS ARE EQUAL, RESULT IS P
0F04 E680              ANI  128             ;MASK OFF UNNEEDED BITS
0F06 B0                ORA  B               ;CREATE SIGN OF QUOTIENT
0F07 12                STAX D               ;PUT TO FACC
0F08 E5                PUSH H               ;SAVE ADDR
0F09 13                INX  D               ;POINT MSD OF DIVIDEND
0F0A 214C20            LXI  H,FTEMP         ;POINT TEMPORARY STORAGE
0F0D 3600              MVI  M,0             ;CLEAR MSB
0F0F 23                INX  H               ;POINT NEXT
0F10 0603              MVI  B,3             ;LOOP CTR
0F12 1A        FDIV3:  LDAX D               ;GET BYTE FROM FACC
0F13 77                MOV  M,A             ;PUT TO FTEMP
0F14 AF                XRA  A               ;CLEAR A
0F15 12                STAX D               ;ZERO FACC
0F16 23                INX  H               ;POINT NEXT
0F17 13                INX  D               ;DITTO
0F18 05                DCR  B               ;DECR CTR
0F19 C2120F            JNZ  FDIV3           ;LOOP
0F1C D1                POP  D               ;GET ADDR
0F1D 0603              MVI  B,3             ;LOOP CTR
0F1F 13                INX  D               ;POINT MSD OF DIVISOR
0F20 3600              MVI  M,0             ;CLEAR MSB
0F22 23                INX  H               ;POINT NEXT
0F23 CD241C            CALL COPYD           ;GO MOVE IT
0F26 0E18              MVI  C,24            ;OUTER LOOP CTR
0F28 114F20    FDIV5:  LXI  D,FTEMP+3       ;POINT DIVIDEND
```



```
ADDR OBJECT    TAG     OPCD OPRNDS          COMMENTS

0F2B 215320            LXI  H,FTEMP+7       ;AND DIVISOR
0F2E 0604              MVI  B,4             ;CTR
0F30 CD8A0F            CALL FSUBT           ;GO SUBTRACT
0F33 D2420F            JNC  FDIV6           ;BRIF NO GO
0F36 114F20            LXI  D,FTEMP+3       ;POINT DIVIDEND
0F39 215320            LXI  H,FTEMP+7       ;AND DIVISOR
0F3C 0604              MVI  B,4             ;CTR
0F3E CD950F            CALL FADDT           ;GO RE-ADD
0F41 37                STC                  ;TURN ON CY
0F42 3F        FDIV6:  CMC                  ;REVERSE CY
0F43 0603              MVI  B,3             ;CTR
0F45 214B20            LXI  H,FACC+3        ;POINT LSB
0F48 7E        FDIV7:  MOV  A,M             ;LOAD BYTE
0F49 17                RAL                  ;SHIFT LEFT
0F4A 77                MOV  M,A             ;REPLACE
0F4B 2B                DCX  H               ;POINT NEXT
0F4C 05                DCR  B               ;DECR CTR
0F4D C2480F            JNZ  FDIV7           ;LOOP
0F50 AF                XRA  A               ;CLEAR FLAGS
0F51 0604              MVI  B,4             ;CTR
0F53 214F20            LXI  H,FTEMP+3       ;POINT DIVIDEND
0F56 7E        FDIV8:  MOV  A,M             ;LOAD BYTE
0F57 17                RAL                  ;SHIFT LEFT
0F58 77                MOV  M,A             ;REPLACE
0F59 2B                DCX  H               ;POINT ENXT
0F5A 05                DCR  B               ;DECR CTR
0F5B C2560F            JNZ  FDIV8           ;LOOP
0F5E 0D                DCR  C               ;DECR OTR CTR
0F5F C2280F            JNZ  FDIV5           ;LOOP
0F62 C31D0E            JMP  FNCRM           ;WRAPUP
```

```
ADDR OBJECT   LINE TAG    OPCD OPRNDS          COMMENTS

0F65               FOVUN: EQU  $
                   ;
                   ;  TEST EXPONENT FOR OVERFLO OR UNDERFLOW
                   ;
0F65 B7                   ORA  A               ;TEST IT
0F66 F26F0F               JP   FOV1            ;BRIF POS.
0F69 FEC1                 CPI  193             ;TEST FOR MAX NEG
0F6B D0                   RNC                  ;RETURN IF NO UNDER.
0F6C C3981A               JMP  OVERR           ;GO ERROR
0F6F FE40          FOV1:  CPI  64              ;TEST MAX POS
0F71 D8                   RC                   ;RETURN IF NO OVER.
0F72 C3981A               JMP  OVERR           ;GO ERROR
```



```
ADDR OBJECT   LINE TAG    OPCD OPRNDS          COMMENTS

0F75               FTEST: EQU  $
                   ;
                   ;  TEST THE SIGN OF THE NUMBER IN THE FACC
                   ;  RETURN WITH S & Z SET TO SIGN
                   ;
0F75 3A4920               LDA  FACC+1          ;GET MSD
0F78 B7                   ORA  A               ;TEST IT
0F79 C8                   RZ                   ;RETURN IF ZERO
0F7A 3A4820               LDA  FACC            ;GET SIGN&EXPON BYTE
0F7D F67F                 ORI  127             ;TEST SIGN BIT ONLY
0F7F 3A4820               LDA  FACC            ;RE-LOAD EXPON BYTE
0F82 C9                   RET                  ;THEN RETURN
```



```
ADDR OBJECT   LINE TAG    OPCD OPRNDS          COMMENTS

0F83               FEXP:  EQU  $
                   ;
                   ;  EXPAND EXPONENT INTO 8 BINARY BITS
                   ;
0F83 E67F                 ANI  127             ;MASK MANTISA SIGN
0F85 C640                 ADI  64              ;PROPAGATE CHAR SIGN INTO LEFTMO
0F87 EE40                 XRI  64              ;RESTORE ORIGINAL SIGN BIT
0F89 C9                   RET                  ;RETURN
```

```
ADDR OBJECT    LINE TAG     OPCD OPRNDS          COMMENTS

0F8A           2528 FSUBT:  EQU  $
               2529 ;
               2530 ; SUBTRACT THE TWO MULTIPRECISION NUMBERS (D,E) & (H,L)
               2531 ;
0F8A AF        2532         XRA  A               ;TURN OFF CY
0F8B 1A        2533 FSB1:   LDAX D               ;GET A BYTE
0F8C 9E        2534         SBB  M               ;SUB OTHER BYTE
0F8D 12        2535         STAX D               ;PUT DOWN
0F8E 1B        2536         DCX  D               ;POINT NEXT
0F8F 2B        2537         DCX  H               ;DITTO
0F90 05        2538         DCR  B               ;DECR CTR
0F91 C28B0F    2539         JNZ  FSB1            ;LOOP
0F94 C9        2540         RET                  ;RETURN
```



```
ADDR OBJECT    LINE TAG     OPCD OPRNDS          COMMENTS

0F95           2542 FADDT:  EQU  $
               2543 ;
               2544 ; ADD TWO MULTI-PRECISION NUMBERS (D,E) & (H,L)
               2545 ;
0F95 AF        2546         XRA  A               ;CLEAR STATUS
0F96 1A        2547 FAD1:   LDAX D               ;GET BYTE
0F97 8E        2548         ADC  M               ;ADD OTHER BYTE
0F98 12        2549         STAX D               ;PUT DOWN
0F99 1B        2550         DCX  D               ;POINT NEXT
0F9A 2B        2551         DCX  H               ;DITTO
0F9B 05        2552         DCR  B               ;DECR LOOP CTR
0F9C C2960F    2553         JNZ  FAD1            ;LOOP
0F9F C9        2554         RET                  ;RETURN
```

```
ADDR OBJECT    LINE TAG     OPCD OPRNDS             COMMENTS

0FA0           2556 FSHFT:  EQU  $
               2557 ;
               2558 ; INCREMENTING SHIFT RIGHT
               2559 ;
0FA0 7E        2560         MOV  A,M                ;GET A BYTE
0FA1 1F        2561         RAR                     ;SHIFT RIGHT
0FA2 77        2562         MOV  M,A                ;PUT DOWN
0FA3 23        2563         INX  H                  ;POINT NEXT
0FA4 05        2564         DCR  B                  ;DECR CTR
0FA5 C2A00F    2565         JNZ  FSHFT              ;LOOP
0FA8 C9        2566         RET                     ;RETURN
```



```
ADDR OBJECT    LINE TAG     OPCD OPRNDS             COMMENTS

0FA9           2568 SIN:    EQU  $
               2569 ;
               2570 ; COMPUTE SINE OF X. (X IN RADIANS)
               2571 ;
               2572 ; USES 4TH DEGREE POLYNOMIAL APPROXIMATION
               2573 ;
               2574 ;
               2575 ; FIRST, REDUCE ANGLE TO RANGE: (-PI/2,PI/2)
               2576 ;
0FA9 AF        2577         XRA  A                  ;INIT SIGN
0FAA F5        2578         PUSH PSW                ;SAVE IT
0FAB CD750F    2579         CALL FTEST              ;GET STATUS OF ANGLE
0FAE CAB70F    2580         JZ   SIN1               ;BRIF ZERO
0FB1 F2BB0F    2581         JP   SIN2               ;BRIF POSITIVE
0FB4 CD2A11    2582         CALL ABS                ;ELSE, GET ABSOLUTE VALUE
0FB7 F1        2583 SIN1:   POP  PSW                ;GET PREV SIGN
0FB8 EE80      2584         XRI  128                ;REVERSE
0FBA F5        2585         PUSH PSW                ;RESAVE
0FBB 21821E    2586 SIN2:   LXI  H,HALFP            ;POINT PI/2
0FBE CD530E    2587         CALL FSUB               ;GO SUBTRACT
0FC1 CAD00F    2588         JZ   SIN3               ;BRIF ZERO
0FC4 FAD00F    2589         JM   SIN3               ;OR NEG
0FC7 21821E    2590         LXI  H,HALFP            ;ELSE, PI/2
0FCA CD530E    2591         CALL FSUB               ;SUBTRACT
0FCD C3B70F    2592         JMP  SIN1               ;LOOP
0FD0 21821E    2593 SIN3:   LXI  H,HALFP            ;POINT PI/2
0FD3 CD590D    2594         CALL FADD               ;GO ADD
0FD6 F1        2595         POP  PSW                ;GET SIGN
0FD7 FCDA11    2596         CM   NEG                ;BRIF REVERSE SIGN NEEDED
               2597 ;
               2598 ; INIT REGISTERS
               2599 ;
0FDA 212420    2600         LXI  H,TEMP1            ;POINT T1
0FDD F7        2601         RST  6                  ;SAVE IT
0FDE 3A4820    2602         LDA  FACC               ;GET SIGN&EXPONENT
0FE1 CD830F    2603         CALL FEXP               ;EXPAND EXPON.
0FE4 F2EA0F    2604         JP   SINSA              ;BRIF POSITIVE
0FE7 FEFD      2605         CPI  253                ;TEST EXPONENT
0FE9 D8        2606         RC                      ;RETURN IF VERY SMALL RADIAN
               2607 ;
               2608 ; ABOVE ROUTINE WILL APPROXIMATE SIN(X) == X FOR X: (-.06,.06
               2609 ;
0FEA 21821E    2610 SINSA:  LXI  H,HALFP            ;POINT PI/2
0FED CDE30E    2611         CALL FDIV               ;COMPUTE X/PI/2
0FF0 212820    2612         LXI  H,TEMP2            ;POINT T2
0FF3 F7        2613         RST  6                  ;STORE IT
0FF4 212820    2614         LXI  H,TEMP2            ;POINT BACK
0FF7 CD6C0E    2615         CALL FMUL               ;COMPUTE SQUARE
0FFA 212C20    2616         LXI  H,TEMP3            ;POINT T3
0FFD F7        2617         RST  6                  ;SAVE IT
0FFE 212820    2618         LXI  H,TEMP2            ;POINT NEW X
```

```
ADDR OBJECT   LINE TAG    OPCD OPRNDS       COMMENTS

1001 CD6C0E               CALL FMUL         ;COMPUTE CUBE
1004 212320               LXI  H,TEMP2      ;POINT FACTOR AREA
1007 F7                   RST  6            ;SAVE IT
1008 0604                 MVI  B,4          ;NUM OF TERMS
100A 21861E               LXI  H,SINCO      ;POINT CONSTANTS
                   ;
                   ; COMPUTE SINE OF ANGLE
                   ;
100D C5            SIN4:  PUSH B            ;SAVE CTR
100E E5                   PUSH H            ;SAVE ADDRESS
100F CD6C0E        SIN5:  CALL FMUL         ;COMPUTE THIS TERM
1012 212420               LXI  H,TEMP1      ;POINT PRIOR SUM
1015 CD590D               CALL FADD         ;ADD
1018 E1                   POP  H            ;GET ADDRESS CONST
1019 C1                   POP  B            ;GET CTRS
101A 05                   DCR  B            ;DECR CTR
101B C8                   RZ                ;RETURN IF HAVE SINE
101C FF                   RST  7            ;ADJUST TO NEXT CONST
101D 04                   DB   4
101E C5                   PUSH B            ;SAVE CONST
101F E5                   PUSH H            ;SAVE ADDR
1020 212420               LXI  H,TEMP1      ;POINT SUM AREA
1023 F7                   RST  6            ;SAVE IT
1024 EF                   RST  5            ;LOAD X FACTOR
1025 CD6C0E               CALL FMUL         ;GO MULTIPLY BY X^2
1028 212320               LXI  H,TEMP2      ;POINT SAVE AREA
102B F7                   RST  6            ;SAVE IT
102C E1                   POP  H            ;GET ADDRESS
102D E5                   PUSH H            ;RE-SAVE
102E C30F10               JMP  SIN5         ;LOOP
```



```
ADDR OBJECT   LINE TAG    OPCD OPRNDS       COMMENTS

1031               COS:   EQU  $
                   ;
                   ;
                   ; COMPUTE COSINE OF ANGLE, X EXPRESSED IN RADIANS
                   ; USES THE TRANSFORMATION: Y = PI/2 +- X AND THEN COMPUTES SI
                   ;
                   ;
1031 21821E               LXI  H,HALFP      ;COMPUTE PI/2 + X
1034 CD590D               CALL FADD         ;GO ADD
1037 C3A90F               JMP  SIN          ;GO COMPUTE SINE
```

```
ADDR OBJECT   LINE TAG    OPCD OPRNDS         COMMENTS

103A               TAN:   EQU  $
                   ;
                   ; COMPUTE TANGENT OF X, IN RADIANS
                   ; USES THE RELATION:
                   ;
                   ;              SIN(X)
                   ; TAN(X) =  -------
                   ;              COS(X)
                   ;
103A 213C20               LXI  H,TEMP7        ;POINT SAVE AREA
103D F7                   RST  6              ;SAVE ANGLE
103E CD3110               CALL COS            ;COMPUTE COS(X)
1041 214820               LXI  H,FACC         ;POINT ANSWER
1044 113C20               LXI  D,TEMP7        ;POINT X
1047 0604                 MVI  B,4            ;LOOP CTR
1049 4E            TAN1:  MOV  C,M            ;GET BYTE
104A 1A                   LDAX D              ;GET SAVE BYTE
104B 77                   MOV  M,A            ;STORE IT
104C 79                   MOV  A,C            ;GET SAVED FACC
104D 12                   STAX D              ;PUT TO TEMP7
104E 23                   INX  H              ;POINT NEXT
104F 13                   INX  D              ;DITTO
1050 05                   DCR  B              ;DECR CTR
1051 C24910               JNZ  TAN1           ;LOOP
1054 CDA90F               CALL SIN            ;COMPUTE SINE
1057 213C20               LXI  H,TEMP7        ;POINT COS
105A CDE30E               CALL FDIV           ;GO DIVIDE
105D C9                   RET                 ;RETURN
```



```
ADDR OBJECT   LINE TAG    OPCD OPRNDS         COMMENTS

105E               ATN:   EQU  $
                   ;
                   ; COMPUTES THE ARCTANGENT OF X
                   ; USES A SEVENTH DEGREE POLYNOMIAL APPROXIMATION
                   ;
105E CD750F               CALL FTEST          ;CHECK SIGN OF ARGUMENT
1061 F26D10               JP   ATN1           ;BRIF POSITIVE
1064 CDDA11               CALL NEG            ;REVERSE SIGN
1067 CD6D10               CALL ATN1           ;GET POSITIVE ATN
106A C3DA11               JMP  NEG            ;MAKE NEG & RETURN
                   ;
106D 213E1E        ATN1:  LXI  H,ONE          ;POINT: 1
1070 CD590D               CALL FADD           ;GO ADD
1073 212420               LXI  H,TEMP1        ;POINT SAVE
1076 F7                   RST  6              ;STORE
1077 21421E               LXI  H,TWO          ;POINT: 2
107A CD530E               CALL FSUB           ;GO SUBTRACT
107D 212420               LXI  H,TEMP1        ;POINT SAVED
1080 CDE30E               CALL FDIV           ;DIVIDE
1083 212820               LXI  H,TEMP2        ;POINT SAVE
1086 F7                   RST  6              ;SAVE (X-1)/(X+1)
1087 212820               LXI  H,TEMP2        ;POINT SAME
108A CD6C0E               CALL FMUL           ;SQUARE IT
108D 212C20               LXI  H,TEMP3        ;POINT SAVE
1090 F7                   RST  6              ;SAVE ((X-1)/(X+1))^2
1091 11621E               LXI  D,QTRPI        ;POINT PI/4
1094 212420               LXI  H,TEMP1        ;POINT SAVE
1097 0604                 MVI  B,4            ;BYTE COUNT
1099 CD241C               CALL COPYD          ;MOVE IT
109C EF                   RST  5              ;LOAD (X-1)/(X+1)
109D 0607                 MVI  B,7            ;TERM COUNT
109F 21661E               LXI  H,ATNCO        ;POINT LIST COEFFICIENTS
10A2 C30D10               JMP  SIN4           ;GO COMPUTE & RETURN
```

```
ADDR OBJECT   LINE TAG    OPCD OPRNDS          COMMENTS

10A5          2724 LOG:   EQU  $
              2725 ;
              2726 ;
              2727 ; COMPUTES THE NATURAL LOGARITHM, LOG(X)
              2728 ; USES A 7TH DEGREE POLYNOMIAL APPROXIMATION
              2729 ;
10A5 CD750F   2730        CALL FTEST           ;TEST THE ARGUMENT
10A8 FA981A   2731        JM   OVERR           ;BRIF NEG
10AB 212820   2732        LXI  H,TEMP2         ;POINT SAVE AREA
10AE F7       2733        RST  6               ;STORE IT
10AF 3A4820   2734        LDA  FACC            ;GET EXPON
10B2 CD830F   2735        CALL FEXP            ;EXPAND TO 8 BITS
10B5 F2C410   2736        JP   LOG1            ;BRIF POSITIVE EXPONENT
10B8 2F       2737        CMA                  ;ELSE, COMPLEMENT
10B9 C602     2738        ADI  2               ;PLUS TWO
10BB CD7012   2739        CALL FDEC            ;CONVERT TO FLOAT POINT
10BE CDDA11   2740        CALL NEG             ;THEN NEGATE
10C1 C3C910   2741        JMP  LOG2            ;GO AROUND
10C4 DE01     2742 LOG1:  SBI  1               ;MINUS ONE
10C6 CD7012   2743        CALL FDEC            ;CONVERT TO FLOATING POINT
10C9 21521E   2744 LOG2:  LXI  H,LN2           ;POINT LOG(2)
10CC CD6C0E   2745        CALL FMUL            ;MULTIPLY
10CF 212420   2746        LXI  H,TEMP1         ;POINT SAVE AREA
10D2 F7       2747        RST  6               ;STORE IT
10D3 EF       2748        RST  5               ;GET ORIG X
10D4 3E01     2749        MVI  A,1             ;GET EXPONENT: 1
10D6 324820   2750        STA  FACC            ;ADJUST TO RANGE (1,2)
10D9 213E1E   2751        LXI  H,ONE           ;POINT 1
10DC CD530F   2752        CALL FSUB            ;SUBTRACT ONE
10DF 212820   2753        LXI  H,TEMP2         ;POINT STORAGE
10E2 F7       2754        RST  6               ;SAVE X
10E3 F7       2755        RST  6               ;SAVE X^N
10E4 0607     2756        MVI  B,7             ;SEVEN TERMS
10E6 21961E   2757        LXI  H,LOGCO         ;POINT COEFFICIENTS
10E9 C30D10   2758        JMP  SIN4            ;APPROXIMATE & RETURN
```



```
ADDR OBJECT   LINE TAG    OPCD OPRNDS          COMMENTS

10EC          2760 EXP:   EQU  $
              2761 ;
              2762 ;
              2763 ; COMPUTES EXPONENTIAL E^X
              2764 ; USES 6TH DEGREE POLYNOMIAL APPROXIMATION
              2765 ;
10EC CD750F   2766        CALL FTEST           ;CHECK SIGN
10EF F20611   2767        JP   EXP1            ;BRIF POSITIVE
10F2 CDDA11   2768        CALL NEG             ;ELSE, REVERSE SIGN
10F5 CD0611   2769        CALL EXP1            ;COMPUTE POSITIVE EXP
10F8 212420   2770        LXI  H,TEMP1         ;POINT SAVE AREA
10FB F7       2771        RST  6               ;STORE IT
10FC 213E1E   2772        LXI  H,ONE           ;POINT 1
10FF EF       2773        RST  5               ;LOAD IT
1100 212420   2774        LXI  H,TEMP1         ;POINT PREV
1103 C3E30E   2775        JMP  FDIV            ;RECRIPRICAL AND RETURN
              2776 ;
1106 212820   2777 EXP1:  LXI  H,TEMP2         ;POINT SAVE AREA
1109 F7       2778        RST  6               ;SAVE X^N
110A F7       2779        RST  6               ;X
110B 213E1E   2780        LXI  H,ONE           ;POINT: 1
110E 112420   2781        LXI  D,TEMP1         ;POINT SAVE AREA
1111 0604     2782        MVI  B,4             ;BYTE COUNT
1113 CD2D1C   2783        CALL COPYH           ;INIT SUM = 1
1116 0606     2784        MVI  B,6             ;GET TERM COUNT
1118 21B21E   2785        LXI  H,EXPCO         ;POINT COEFFICIENTS
111B CD0D10   2786        CALL SIN4            ;GO COMPUTE APPROX.
111E 214820   2787        LXI  H,FACC          ;POINT PARTIAL ANSWER
1121 CD6C0E   2788        CALL FMUL            ;SQUARE IT
1124 214820   2789        LXI  H,FACC          ;POINT AGAIN
1127 C36C0E   2790        JMP  FMUL            ;^4 AND RETURN
```

```
ADDR OBJECT    TAG    OPCD OPRNDS            COMMENTS

112A           ABS:   EQU  $
               ;
               ;
               ; RETURN THE ABSOLUTE VALUE OF THE FLOATING ACCUMULATOR
               ;
               ;
112A 3A4820           LDA  FACC              ;GET EXPONENT
112D E67F             ANI  127               ;STRIP NEGATIVE SIGN
112F 324820           STA  FACC              ;REPLACE
1132 C9               RET                    ;RETURN
```



```
ADDR OBJECT    TAG    OPCD OPRNDS            COMMENTS

1133           SGN:   EQU  $
               ;
               ;
               ; RETURNS THE SIGN OF THE FLOATING ACCUMULATOR
               ; THAT IS:
               ;  1 IF FACC > 0
               ;  0 IF FACC = 0
               ; -1 IF FACC < 0
               ;
1133 CD750F           CALL FTEST             ;GET STATUS OF FACC
1136 C8               RZ                     ;RETURN IF ZERO
1137 E680             ANI  128               ;ISOLATE SIGN
1139 F601             ORI  1                 ;CREATE EXPONENT
113B F5               PUSH PSW               ;SAVE IT
113C 213E1E           LXI  H,ONE             ;GET ADDRESS OF CONSTANT 1
113F EF               RST  5                 ;GO LOAD IT
1140 F1               POP  PSW               ;RESTORE SIGN
1141 324820           STA  FACC              ;SET THE SIGN
1144 C9               RET                    ;RETURN
```

```
ADDR OBJECT  TAG     OPCD OPRNDS              COMMENTS

1145         INT:    EQU  $
             ;
             ;
             ; RETURNS THE GREATEST INTEGER NOT LARGER THAN VALUE IN FACC
             ; E.G.:
             ;       INT(3.14159) = 3
             ;       INT(0)       = 0
             ;       INT(-3.1415) =-4
             ;
             ;
1145 214820           LXI  H,FACC             ;POINT FLOAT ACC
1148 7E               MOV  A,M                ;GET EXPONENT
1149 E640             ANI  64                 ;GET SIGN OF CHARACTERISTIC
114B CA5311           JZ   INT2               ;BRIF GE ZERO
114E 0604             MVI  B,4                ;LOOP CTR
1150 C3331C           JMP  ZEROM              ;GO ZERO THE FACC
1153 7E      INT2:    MOV  A,M                ;GET EXPONENT AGAIN
1154 B7               ORA  A                  ;TEST SIGN
1155 F26211           JP   INT3               ;BRIF POSITIVE OR ZERO
1158 214E1E           LXI  H,NEGON            ;POINT CONSTANT: - 9999999
115B CD5900           CALL FADD               ;ADD TO FACC
115E 214820           LXI  H,FACC             ;POINT EXPONTENT AGAIN
1161 7E               MOV  A,M                ;LOAD IT
1162 E63F    INT3:    ANI  63                 ;ISOLATE CHARACTERISTIC
1164 FE18             CPI  24                 ;TEST IF ANY FRACTION
1166 F0               RP                      ;RETURN IF NOT
1167 47               MOV  B,A                ;SAVE EXPONENT
1168 3E18             MVI  A,24               ;GET CONSTANT
116A 90               SUB  B                  ;MINUS EXPONENT = LOOP CTR
116B 4F               MOV  C,A                ;SAVE IT
116C 214920  INT4:    LXI  H,FACC+1           ;POINT MSB
116F AF               XRA  A                  ;CLEAR CY FLAG
1170 0603             MVI  B,3                ;BYTE COUNT
1172 7E      INT5:    MOV  A,M                ;LOAD A BYTE
1173 1F               RAR                     ;SHIFT RIGHT
1174 77               MOV  M,A                ;REPLACE
1175 23               INX  H                  ;POINT NEXT
1176 05               DCR  B                  ;DECR BYTE CTR
1177 C27211           JNZ  INT5               ;LOOP
117A 0D               DCR  C                  ;DECR BIT CTR
117B C26C11           JNZ  INT4               ;LOOP
117E 214820           LXI  H,FACC             ;POINT SIGN & EXP
1181 7E               MOV  A,M                ;LOAD IT
1182 E680             ANI  128                ;ISOLATE SIGN
1184 C618             ADI  24                 ;PLUS INTEGER
1186 77               MOV  M,A                ;REPLACE IT
1187 C31D0E           JMP  FNORM              ;GO NORMALIZE & RETURN
```

```
ADDR OBJECT   LINE TAG     OPCD OPRNDS            COMMENTS

113A               SQR:    EQU  $
                   ;
                   ; COMPUTE SQUARE ROOT OF ARUGMENT IN FACC. LEAVE RESULT IN FA
                   ;
                   ; USE HERON'S ITERATIVE PROCESS -
                   ;
118A CD750F                CALL FTEST             ;TEST THE ARGUMENT
118D C8                    RZ                     ;RETURN IF ZERO
118E FA981A                JM   OVERR             ;ERROR IF NEGATIVE
1191 326A21                STA  DEXP              ;SAVE ORIG EXPONENT
1194 AF                    XRA  A                 ;GET A ZERO
1195 324820                STA  FACC              ;PUT ARG IN RANGE [.5, 1]
1198 212820                LXI  H,TEMP2           ;POINT SAVE AREA
119B F7                    RST  6                 ;STORE IT
                   ;
                   ; INITIAL APROXIMATION: 0.41730759 + 0.59016206 * MANTISSA
                   ;
119C 21561E                LXI  H,SQC1            ;POINT .59016
119F CD6C0E                CALL FMUL              ;GO MULTIPLY
11A2 215A1E                LXI  H,SQC2            ;PINT .4173
11A5 CD590D                CALL FADD              ;GO ADD
11A8 212420                LXI  H,TEMP1           ;POINT SAVE AREA
11AB F7                    RST  6                 ;GO STORE IT
                   ;
                   ; NEWTON'S METHOD OF ITERATION TO THE APPROXIMATE
                   ; VALUE OF THE SQR OF MANTISSA
                   ;
11AC CDC411                CALL SQR1              ;FIRST ITERATION
11AF 212420                LXI  H,TEMP1           ;POINT SAVE AREA
11B2 F7                    RST  6                 ;STORE IT
11B3 CDC411                CALL SQR1              ;SECOND ITERATION
                   ;
                   ; RESTORE RANGE TO OBTAIN THE FINAL RESULT
                   ;
11B6 3A6A21                LDA  DEXP              ;GET SAVED EXPONENT
11B9 1F                    RAR                    ;DIVIDE BY 2
11BA 324820                STA  FACC              ;STORE IT
11BD D0                    RNC                    ;RETURN IF EXPON EVEN
11BE 215E1E                LXI  H,SQC3            ;ELSE, POINT SQR(2)
11C1 C36C0E                JMP  FMUL              ;GO MULTIPLY AND RETURN
                   ;
                   ; THIS ROUTINE PERFORMS ONE NEWTON ITERATION
                   ; TO THE SQUARE ROOT FUNCTION
                   ;
11C4 212820        SQR1:   LXI  H,TEMP2           ;POINT MANTISSA
11C7 EF                    RST  5                 ;LOAD IT
11C8 212420                LXI  H,TEMP1           ;POINT PREV GUESS
11CB CDE30E                CALL FDIV              ;FORM MANT/TEMP1
11CE 212420                LXI  H,TEMP1           ;POINT PREV
11D1 CD590D                CALL FADD              ;FORM TEMP1 + MANT/TEMP1
11D4 D601                  SUI  1                 ;DIVIDE BY 2
```



```
ADDR OBJECT   LINE TAG     OPCD OPRNDS            COMMENTS

11D6 324820                STA  FACC              ;FORM (TEMP1 + MANT/TEMP1)/2
11D9 C9                    RET                    ;RETURN
```

```
IMSAI 8080 8K BASIC VER=1.3                          11:03:55  15 APR 1976  PAGE 91

ADDR OBJECT   LINE TAG    OPCD OPRNDS            COMMENTS

11DA          2925 NEG:   EQU  $
              2926 ;
              2927 ;
              2928 ;  REVERSES THE SIGN OF THE FLOATING ACC
              2929 ;
              2930 ;
11DA CD750F   2931        CALL FTEST             ;GET STATUS OF FACC
11DD C8       2932        RZ                     ;RETURN IF ZERO
11DE EE80     2933        XRI  128               ;REVERSE SIGN
11E0 324820   2934        STA  FACC              ;RESTORE EXPONENT
11E3 C9       2935        RET                    ;CONTINUE EVALUATION
```

```
IMSAI 8080 8K BASIC VER=1.3                          11:04:07  15 APR 1976  PAGE 92

ADDR OBJECT   LINE TAG    OPCD OPRNDS            COMMENTS

11E4          2937 RND:   EQU  $
              2938 ;
              2939 ;
              2940 ;  PSEUDO RANDOM NUMBER GENERATOR
              2941 ;
              2942 ;
11E4 0604     2943        MVI  B,4               ;LOOP CTR
11E6 214820   2944        LXI  H,FACC            ;POINT FLOAT ACCUM
11E9 CD331C   2945        CALL ZEROM             ;GO ZERO THE FACC
11EC 0E03     2946        MVI  C,3               ;OUTTER LOOP CTR
11EE 214920   2947        LXI  H,FACC+1          ;POINT MSB
11F1 E5       2948        PUSH H                 ;SAVE H,L
11F2 217921   2949 RND1:  LXI  H,RNDZ+1          ;POINT X,Y,Z
11F5 0606     2950        MVI  B,6               ;LOOP CTR
11F7 B7       2951        ORA  A                 ;TURN OFF CY
11F8 7E       2952 RND2:  MOV  A,M               ;GET A BYTE
11F9 17       2953        RAL                    ;SHIFT LEFT (MULT BY 2)
11FA 77       2954        MOV  M,A               ;REPLACE THE BYTE
11FB 2B       2955        DCX  H                 ;POINT NEXT
11FC 05       2956        DCR  B                 ;DECR CTR
11FD C2F811   2957        JNZ  RND2              ;LOOP
1200 23       2958        INX  H                 ;POINT MSD X,Y,Z
1201 11561D   2959        LXI  D,RNDP            ;POINT TO MODULO
1204 0603     2960        MVI  B,3               ;LOOP CTR
1206 1A       2961 RND3:  LDAX D                 ;GET BYTE OF P,Q,R
1207 BE       2962        CMP  M                 ;COMPARE WITH X,Y,Z
1208 13       2963        INX  D                 ;POINT NEXT
1209 23       2964        INX  H                 ;DITTO
120A DA1512   2965        JC   RND4              ;BRIF P<X
120D C22112   2966        JNZ  RND5              ;BRIF P>X
1210 1A       2967        LDAX D                 ;GET LOW BYTE
1211 BE       2968        CMP  M                 ;CMPARE
1212 D22112   2969        JNC  RND5              ;BRIF P>=X
1215 EB       2970 RND4:  XCHG                   ;FLIP D,E TO H,L
1216 1A       2971        LDAX D                 ;GET LOW X BYTE
1217 96       2972        SUB  M                 ;SUBTRACT LOW P BYTE
1218 12       2973        STAX D                 ;STORE IT
1219 1B       2974        DCX  D                 ;POINT HIGH
121A 2B       2975        DCX  H                 ;DITTO
121B 1A       2976        LDAX D                 ;GET HIGH X BYTE
121C 9E       2977        SBB  M                 ;SUB HIGH P BYTE
121D 12       2978        STAX D                 ;STORE IT
121E 13       2979        INX  D                 ;POINT LOW
121F 23       2980        INX  H                 ;DITTO
1220 EB       2981        XCHG                   ;RESTORE ADDRS
1221 13       2982 RND5:  INX  D                 ;POINT NEXT
1222 23       2983        INX  H                 ;DITTO
1223 05       2984        DCR  B                 ;DECR CTR
1224 C20612   2985        JNZ  RND3              ;LOOP
1227 0603     2986        MVI  B,3               ;LOOP CTR
1229 117B21   2987 RND6:  LXI  D,RNDS+1          ;POINT LOW S
```

```
ADDR OBJECT   LINE TAG     OPCD OPRNDS          COMMENTS

122C 1A       2988         LDAX D               ;GET LOW S
122D 86       2989         ADD  M               ;ADD LOW X,Y,Z
122E 12       2990         STAX D               ;PUT S
122F 1B       2991         DCX  D               ;POINT HIGH
1230 2B       2992         DCX  H               ;DITTO
1231 1A       2993         LDAX D               ;GET HIGH S
1232 8E       2994         ADC  M               ;ADD HIGH X,Y,Z
1233 E63F     2995         ANI  63              ;TURN OFF HIGH BITS
1235 12       2996         STAX D               ;STORE IT
1236 2B       2997         DCX  H               ;POINT NEXT X,Y,Z
1237 05       2998         DCR  B               ;DECR CTR
1238 C22912   2999         JNZ  RND6            ;LOOP
123B 3E08     3000         MVI  A,8             ;CONSTANT
123D 91       3001         SUB  C               ;LESS CTR
123E 1F       3002         RAR                  ;DIVIDE BY TWO
123F E1       3003         POP  H               ;GET H,L ADDR
1240 3A7B21   3004         LDA  RNDS+1          ;GET LSB OF S
1243 77       3005         MOV  M,A             ;STORE IT
1244 23       3006         INX  H               ;POINT NEXT
1245 E5       3007         PUSH H               ;SAVE H,L
1246 0D       3008         DCR  C               ;DECR CTR
1247 C2F211   3009         JNZ  RND1            ;LOOP
124A E1       3010         POP  H               ;RESTORE SP PTR
124B 3A7C21   3011         LDA  RNDSW           ;GET SWITCH
124E B7       3012         ORA  A               ;TEST IT
124F CA1D0E   3013         JZ   FNORM           ;BRIF NO RANDOMIZE
1252 116C21   3014         LXI  D,TRNDX         ;POINT SAVED VALUES
1255 217421   3015         LXI  H,RNDX          ;POINT NEXT VALUES
1258 0608     3016         MVI  B,8             ;LOOP CTR
125A CD2D1C   3017         CALL COPYH           ;GO COPY
125D C31D0E   3018         JMP  FNORM           ;GO NORMALIZE THE NUMBER
```



```
ADDR OBJECT   LINE TAG     OPCD OPRNDS          COMMENTS

1260          3020 INP:    EQU  $
              3021 ;
              3022 ;
              3023 ;  INPUT A BYTE FROM THE DEVICE IN FACC
              3024 ;
              3025 ;  PUT THE RESULT IN THE FACC
              3026 ;
1260 CD3B1C   3027         CALL FBIN            ;CONVERT FACC TO BINARY
1263 210F20   3028         LXI  H,OUTA          ;POINT INSTR BUFFER
1266 36DB     3029         MVI  M,219           ;IN INSTR
1268 23       3030         INX  H               ;POINT NEXT
1269 77       3031         MOV  M,A             ;MOVE ADDR
126A 23       3032         INX  H               ;POINT NEXT
126B 36C9     3033         MVI  M,201           ;RET INSTR
126D CD0F20   3034         CALL OUTA            ;GO INPUT A BYTE
1270 5F       3035 FDEC:   MOV  E,A             ;MOVE BYTE TO LO D,E
1271 1600     3036         MVI  D,0             ;ZERO HI D,E
1273 C3471D   3037         JMP  BINFL           ;GO CONVERT TO DEC & RET
```



```
ADDR OBJECT   LINE TAG     OPCD OPRNDS          COMMENTS

1276          3039 POS:    EQU  $
              3040 ;
              3041 ;
              3042 ;  RETURNS THE CURRENT POSITION OF THE TTY CURSOR
              3043 ;
              3044 ;
1276 3A6B21   3045         LDA  COLUM           ;GET POSITION
1279 C37012   3046         JMP  FDEC            ;CONVERT TO FLOAT AND RETURN
```

```
ADDR OBJECT   LINE TAG     OPCD OPRNDS          COMMENTS

127C               CONCA:  EQU  $
                   ;
                   ;
                   ;  CONCATONATE TWO STRINGS TOGETHER
                   ;  COMBINED LENGTH <= 255
                   ;
127C D1                    POP  D               ;ADJUST STACK
127D 115820                LXI  D,STRIN         ;POINT STRING BUFFER
1280 1A                    LDAX D               ;GET CURRENT LENGTH
1281 4F                    MOV  C,A             ;STORE IT
1282 0600                  MVI  B,0             ;CLEAR HI
1284 EB                    XCHG                 ;FLIP FLOP
1285 09                    DAD  B               ;COMPUTE NEXT
1286 EB                    XCHG                 ;FLIP BACK
1287 86                    ADD  M               ;COMPUTE COMBINED LENGTH
1288 46                    MOV  B,M             ;SAVE LEN2
1289 D29212                JNC  CONC2           ;BRIF NO OVFLW
128C 3EFF                  MVI  A,255           ;MAX LEN
128E 91                    SUB  C               ;MINUS 1ST PART
128F 47                    MOV  B,A             ;SAVE LEN
1290 3EFF                  MVI  A,255           ;UPDATED LENGTH
1292 325820        CONC2:  STA  STRIN           ;STORE IT
1295 78                    MOV  A,B             ;GET LEN TO MOVE
1296 B7                    ORA  A               ;TEST IT
1297 CAA212                JZ   CONC4           ;BRIF NULL
129A 23            CONC3:  INX  H               ;POINT NEXT
129B 13                    INX  D               ;DITTO
129C 7E                    MOV  A,M             ;GET NEXT CHAR
129D 12                    STAX D               ;PUT IT
129E 05                    DCR  B               ;DECR COUNT
129F C29A12                JNZ  CONC3           ;LOOP
12A2 E1            CONC4:  POP  H               ;GET H,L
12A3 2B                    DCX  H               ;POINT BACK
12A4 3A5820                LDA  STRIN           ;GET LEN
12A7 1F                    RAR                  ;DIVIDE BY TWO
12A8 3C                    INR  A               ;PLUS ONE
12A9 EB                    XCHG                 ;SAVE H,L
12AA 2A5921                LHLD SPCTR           ;GET CTR
12AD 4F                    MOV  C,A             ;SAVE CTR
12AE 0600                  MVI  B,0             ;ZERO HI BYTE
12B0 09                    DAD  B               ;ADD LEN THIS STRNG
12B1 225921                SHLD SPCTR           ;SAVE CTR
12B4 C1                    POP  B
12B5 210000                LXI  H,0             ;GET ADDR ZERO
12B8 E5            CONC5:  PUSH H               ;2 BYTE WORD
12B9 3D                    DCR  A               ;DECR CTR
12BA C2B812                JNZ  CONC5           ;CONTINUE
12BD 39                    DAD  SP              ;GET ADDRESS IN H,L
12BE EB                    XCHG                 ;PUT STACK PTR IN D,E
12BF 72                    MOV  M,D             ;MOVE HI ADDR
12C0 23                    INX  H               ;POINT NEXT
```



```
ADDR OBJECT   LINE TAG     OPCD OPRNDS          COMMENTS

12C1 73                    MOV  M,E             ;MOVE LO ADDR
12C2 23                    INX  H               ;POINT NEXT
12C3 36E7                  MVI  M,231           ;TYPE=STRING
12C5 E5                    PUSH H               ;SAVE H,L
12C6 215820                LXI  H,STRIN         ;GET TEMP STR
12C9 7E                    MOV  A,M             ;GET LENGTH
12CA 3C                    INR  A               ;PLUS ONE
12CB 4F                    MOV  C,A             ;SAVE IT
12CC 7E            CONC6:  MOV  A,M             ;GET A BYTE
12CD 12                    STAX D               ;PUT IT DOWN
12CE 13                    INX  D               ;POINT NEXT
12CF 23                    INX  H               ;DITTO
12D0 0D                    DCR  C               ;SUBT CTR
12D1 C2CC12                JNZ  CONC6           ;LOOP
12D4 E1                    POP  H               ;RESTORE H,L
12D5 FF                    RST  7               ;ADJUST H,L
12D6 F9                    DB   -7
12D7 3E04                  MVI  A,4             ;DELETE 4 BYTES
12D9 CDED1A                CALL SQUIS           ;GO COMPRESS
12DC C35E16                JMP  EVAL            ;CONTINUE EVALUATION
```

```
ADDR OBJECT   LINE TAG    OPCD OPRNDS          COMMENTS

12DF               LEN#:   EQU  $
                   ;
                   ;  X=LEN(A$)
                   ;
                   ;  RETURN THE LENGTH OF THE STRING
                   ;
12DF 3A5820                LDA  STRIN          ;GET LEN IN ACC
12E2 C37012                JMP  FDEC           ;GO CONVERT TO DECIMAL & RETURN
```



```
ADDR OBJECT   LINE TAG    OPCD OPRNDS          COMMENTS

12E5               CHR$:   EQU  $
                   ;
                   ;  A$=CHR$(X)
                   ;
                   ;  RETURNS A ONE CHAR STRING HAVING THE ASCII VALUE - X
                   ;
12E5 CD3B1C                CALL FBIN           ;CONVERT FACC TO BINARY
12E8 215820                LXI  H,STRIN        ;POINT OUT AREA
12EB 3601                  MVI  M,1            ;LEN=1
12ED 23                    INX  H              ;POINT NEXT
12EE 77                    MOV  M,A            ;STORE THE CHAR
12EF C9                    RET                 ;RETURN
```

```
IMSAI 8080 8K BASIC VER=1.3                              11:10:36  15 APR 1976  PAGE 100

ADDR OBJECT   LINE TAG    OPCD OPRNDS              COMMENTS

12F0          3142 ASCII: EQU  $
              3143 ;
              3144 ;  X=ASCII(A$)
              3145 ;
              3146 ;  RETURNS THE ASCII VALUE OF THE FIRST CHAR IN THE STRING
              3147 ;
12F0 215820   3148        LXI  H,STRIN             ;POINT STRING
12F3 7E       3149        MOV  A,M                 ;GET LENGTH
12F4 B7       3150        ORA  A                   ;TEST IF > ZERO
12F5 CA7012   3151        JZ   FDEC                ;BRIF ZERO & RETURN A ZERO
12F8 23       3152        INX  H                   ;POINT 1ST CHAR
12F9 7E       3153        MOV  A,M                 ;LOAD IT
12FA C37012   3154        JMP  FDEC                ;GO CONVERT TO DECIMAL & RETURN



IMSAI 8080 8K BASIC VER=1.3                              11:10:55  15 APR 1976  PAGE 101

ADDR OBJECT   LINE TAG    OPCD OPRNDS              COMMENTS

12FD          3156 NUM$:  EQU  $
              3157 ;
              3158 ;  A$=NUM$(X)
              3159 ;
              3160 ;  RETURNS A STRING REPRESENTING X AS IT WOULD HAVE
              3161 ;  BEEN PRINTED (INCLUDING TRAILING SPACE)
              3162 ;
12FD 215820   3163        LXI  H,STRIN             ;POINT STRING AREA
1300 3600     3164        MVI  M,0                 ;INIT COUNT
1302 23       3165        INX  H                   ;SKIP TO 1ST POSITION
1303 CDF30B   3166        CALL FOUT                ;GO CONVERT TO EXTRN DEC
1306 AF       3167        XRA  A                   ;GET A ZERO
1307 47       3168        MOV  B,A                 ;INIT CTR
1308 2B       3169 NUM1:  DCX  H                   ;POINT PRIOR
1309 04       3170        INR  B                   ;COUNT IT
130A BE       3171        CMP  M                   ;TEST IF ZERO
130B C20813   3172        JNZ  NUM1                ;LOOP TILL AT START
130E 70       3173        MOV  M,B                 ;SET LEN CODE
130F C9       3174        RET                      ;THEN RETURN
```

```
ADDR OBJECT   LINE TAG    OPCD OPRNDS          COMMENTS

1310          3176 VAL:   EQU  $
              3177 ;
              3178 ;  X=VAL(A$)
              3179 ;
              3180 ;  RETURNS THE VALUE OF THE STRING OF NUMERIC CHARACTERS
              3181 ;
1310 215820   3182        LXI  H,STRIN        ;POINT STRING AREA
1313 7E       3183        MOV  A,M            ;GET LEN
1314 B7       3184        ORA  A              ;TEST FOR NULL STRING
1315 47       3185        MOV  B,A            ;SAVE LEN
1316 CA7012   3186        JZ   FDEC           ;BRIF IS (RETURNS A 0.00)
1319 115820   3187        LXI  D,STRIN        ;POINT BUFFER
131C 23       3188 VAL1:  INX  H              ;POINT NEXT
131D 7E       3189        MOV  A,M            ;GET A CHAR
131E FE20     3190        CPI  ' '            ;TEST IF SPACE
1320 CA2513   3191        JZ   VAL2           ;BRIF IS
1323 12       3192        STAX D              ;PUT THE CHAR
1324 13       3193        INX  D              ;INCR ADDR
1325 05       3194 VAL2:  DCR  B              ;DECR CTR
1326 C21C13   3195        JNZ  VAL1           ;LOOP
1329 AF       3196        XRA  A              ;GET A ZERO
132A 12       3197        STAX D              ;PUT IN BUFF
132B 215820   3198        LXI  H,STRIN        ;POINT START OF BUFFER
132E CD1B0B   3199        CALL FIN            ;GO CONVERT
1331 7E       3200        MOV  A,M            ;GET NON-NUMERIC
1332 B7       3201        ORA  A              ;TEST IT
1333 C2B01A   3202        JNZ  CVERR          ;BRIF ERROR
1336 C9       3203        RET                 ;ELSE, RETURN
```



```
ADDR OBJECT   LINE TAG    OPCD OPRNDS          COMMENTS

1337          3205 SPACE: EQU  $
              3206 ;
              3207 ;  A$=SPACE$(X)
              3208 ;
              3209 ;  CREATES A STRING OF SPACES LENGTH = X
              3210 ;
1337 CD3B1C   3211        CALL FBIN           ;GET BINARY LENGTH
133A 215820   3212        LXI  H,STRIN        ;POINT TEMP STRING
133D 77       3213        MOV  M,A            ;PUT LEN
133E B7       3214        ORA  A              ;TEST IT
133F C8       3215 SPAC1: RZ                  ;RETURN IF ZERO
1340 23       3216        INX  H              ;ELSE, POINT NEXT
1341 3620     3217        MVI  M,' '          ;MOVE 1 SPACE
1343 3D       3218        DCR  A              ;DECR CTR
1344 C33F13   3219        JMP  SPAC1          ;LOOP
```

```
ADDR OBJECT   LINE TAG    OPCD OPRNDS          COMMENTS

1347          3221 STR$:  EQU  $
              3222 ;
              3223 ; A$=STRING$(X,Y)
              3224 ;
              3225 ; CREATES A STRING OF LENGTH X CONTAINING REPETITION OF CHR$(
              3226 ;
1347 CD3B1C   3227        CALL FBIN            ;GET BINARY LENGTH
134A 325820   3228        STA  STRIN           ;PUT TO STRING
134D CD721C   3229        CALL ARGNU           ;GET NEXT ARGUMENT
1350 215820   3230        LXI  H,STRIN         ;POINT STRING
1353 46       3231        MOV  B,M             ;GET COUNT
1354 23       3232 STRI1: INX  H               ;POINT NEXT
1355 77       3233        MOV  M,A             ;STORE THE CHAR
1356 05       3234        DCR  B               ;DECR CTR
1357 C25413   3235        JNZ  STRI1           ;LOOP
135A C9       3236        RET                  ;RETURN
```



```
ADDR OBJECT   LINE TAG    OPCD OPRNDS          COMMENTS

135B          3238 LEFT$: EQU  $
              3239 ;
              3240 ; B$=LEFT$(A$,X)
              3241 ;
              3242 ; SUBSTRING FROM THE LEFTMOST X CHARACTERS OF A$
              3243 ;
135B CD721C   3244        CALL ARGNU           ;GET 2ND ARGUMENT
135E 4F       3245        MOV  C,A             ;SAVE LEN
135F 0601     3246        MVI  B,1             ;INIT START
1361 C37713   3247        JMP  MIDO            ;CONTINUE
```

```
ADDR OBJECT    LINE TAG     OPCD OPRNDS             COMMENTS

1364                RGHT$:  EQU  $
                    ;
                    ;  B$=RIGHT$(A$,X)
                    ;
                    ;  SUBSTRING STARTING AT POSITION X TO END OF STRING
                    ;
1364 CD721C                 CALL ARGNU             ;GET 2ND ARGUMENT
1367 47                     MOV  B,A               ;SAVE START
1368 0EFF                   MVI  C,255             ;MAX LEN
136A C37713                 JMP  MID0              ;CONTINUE
```



```
ADDR OBJECT    LINE TAG     OPCD OPRNDS             COMMENTS

136D                MID$:   EQU  $
                    ;
                    ;  B$=MID$(A$,X,Y)
                    ;
                    ;  SUBSTRING OF THE STRING A$ STARTING WITH CHARACTER NUMBER X
                    ;  AND Y CHARACTERS LONG
                    ;
136D CD721C                 CALL ARGNU             ;LOAD X
1370 47                     MOV  B,A               ;SAVE START
1371 C5                     PUSH B                 ;PUT ON STACK
1372 CD721C                 CALL ARGNU             ;GET 3RD ARG
1375 C1                     POP  B                 ;RETREIVE
1376 4F                     MOV  C,A               ;SAVE LEN
1377 78             MID0:   MOV  A,B               ;LOAD START
1378 215820                 LXI  H,STRIN           ;POINT STRING
137B BE                     CMP  M                 ;TEST IF X>L
137C DA8513                 JC   MID1              ;BRIF X>L
137F CA8513                 JZ   MID1              ;OR EQUAL
1382 3600                   MVI  M,0               ;ELSE, RESULT IS NULL
1384 C9                     RET                    ;RETURN
1385 81             MID1:   ADD  C                 ;COMPUTE END POSITION
1386 DA9213                 JC   MID2              ;BRIF OVERFLOW
1389 DE01                   SBI  1                 ;COMPUTE X+Y-1
138B DA9213                 JC   MID2              ;BRIF OVERFLOW
138E BE                     CMP  M                 ;COMPARE TO EXISTING LEN
138F DA9613                 JC   MID3              ;BRIF X+Y-1<LEN(A$)
1392 7E             MID2:   MOV  A,M               ;ELSE GET ORIG LEN
1393 90                     SUB  B                 ;MINUS X
1394 3C                     INR  A                 ;PLUS ONE
1395 4F                     MOV  C,A               ;SAVE (REPLACE Y)
1396 71             MID3:   MOV  M,C               ;PUT NEW LEN
1397 58                     MOV  E,B               ;PUT START IN LO
1398 1600                   MVI  D,0               ;ZERO IN HI
139A 19                     DAD  D                 ;COMPUTE START
139B 115820                 LXI  D,STRIN           ;GET BEGIN
139E 7E             MID4:   MOV  A,M               ;GET A CHAR
139F 13                     INX  D                 ;POINT NEXT
13A0 23                     INX  H                 ;DITTO
13A1 12                     STAX D                 ;PUT DOWN
13A2 0D                     DCR  C                 ;DECR CTR
13A3 C29E13                 JNZ  MID4              ;LOOP
13A6 C9                     RET                    ;THEN RETURN
```

```
ADDR OBJECT   LINE TAG     OPCD OPRNDS          COMMENTS

13A7                INSTR: EQU  $
                    ;
                    ;  X=INSTR(Y,A$,B$)
                    ;
                    ;  SEARCH FOR SUBSTRING B$ WITHIN STRING A$ STARTING AT POSITI
                    ;  RETURN 0 IF B$ IS NOT IN A$
                    ;  RETURN 1 IF B$ IS NULL
                    ;  ELSE RETURN THE CHARACTER POSITION
                    ;
13A7 CD721C                CALL ARGNU           ;GET A$
13AA 215820                LXI  H,STRIN         ;POINT A$
13AD B7                    ORA  A               ;TEST Y
13AE C2B613                JNZ  INST2           ;BRIF Y NOT ZERO
13B1 3600           INST1: MVI  M,0             ;ELSE A$ IS NULL
13B3 C3BD13                JMP  INST3           ;GO AROUND
13B6 BE             INST2: CMP  M               ;TEST Y TO LEN(A$)
13B7 CABD13                JZ   INST3           ;BRIF EQUAL
13BA D2B113                JNC  INST1           ;BRIF Y > LEN(A$)
13BD 4F             INST3: MOV  C,A             ;SAVE Y
13BE 0600                  MVI  B,0             ;ZERO HI INCR
13C0 7E                    MOV  A,M             ;GET LEN(A$)
13C1 91                    SUB  C               ;MINUS Y
13C2 3C                    INR  A               ;PLUS ONE
13C3 09                    DAD  B               ;COMPUTE START ADDR
13C4 47                    MOV  B,A             ;# CHARS REMAIN IN A$
13C5 E5             INST4: PUSH H               ;SAVE ADDR
13C6 2A1220                LHLD ADDR1           ;GET ADDR OF ARG
13C9 23                    INX  H               ;POINT NEXT
13CA 56                    MOV  D,M             ;GET HI ADDR
13CB 23                    INX  H               ;POINT NEXT
13CC 5E                    MOV  E,M             ;GET LO ADDR
13CD 23                    INX  H               ;POINT NEXT
13CE 221220                SHLD ADDR1           ;UPDATED PTR
13D1 E1                    POP  H               ;RESTORE ADDR
13D2 1A                    LDAX D               ;GET LEN(B$)
13D3 B7                    ORA  A               ;TEST IF NULL
13D4 C2DD13                JNZ  INST6           ;BRIF NOT
13D7 0E01                  MVI  C,1             ;SET POSIT = 1
13D9 79             INST5: MOV  A,C             ;GET POSIT
13DA C37012                JMP  FDEC            ;CONVERT TO DECIMAL & RETURN
13DD EB             INST6: XCHG                 ;FLIP/FLOP
13DE 78                    MOV  A,B             ;GET LEN OF A$
13DF BE                    CMP  M               ;COMPARE TO LEN B$
13E0 DA0214                JC   INSTA           ;BRIF LEN(B$)< LEN(REM A$)
13E3 C5             INST7: PUSH B               ;SAVE CTR, POSIT
13E4 D5                    PUSH D               ;SAVE ADDR A$
13E5 E5                    PUSH H               ;SAVE ADDR B$
13E6 4E                    MOV  C,M             ;GET LEN B$
13E7 EB                    XCHG                 ;FLIP/FLOP
13E8 13             INST8: INX  D               ;POINT NEXT B$
13E9 1A                    LDAX D               ;GET B$ CHAR
```



```
ADDR OBJECT   LINE TAG     OPCD OPRNDS          COMMENTS

13EA BE                    CMP  M               ;COMPARE A$ CHAR
13EB C2F913                JNZ  INST9           ;BRIF NOT EQUAL
13EE 23                    INX  H               ;POINT NEXT A$
13EF 0D                    DCR  C               ;DECR CTR (LEN(B$))
13F0 C2E813                JNZ  INST8           ;LOOP
13F3 E1                    POP  H               ;DUMMY POP
13F4 E1                    POP  H               ;GET DUMMY STACK
13F5 C1                    POP  B               ;GET POSITION
13F6 C3D913                JMP  INST5           ;WE FOUND A MATCH
13F9 D1             INST9: POP  D               ;GET PTR B$
13FA E1                    POP  H               ;GET PTR A$
13FB C1                    POP  B               ;GET CTRS, POSIT
13FC 0C                    INR  C               ;UP PTR NUM
13FD 23                    INX  H               ;POINT NEXT A$
13FE 05                    DCR  B               ;DECR B
13FF C2DD13                JNZ  INST6           ;LOOP
1402 0E00           INSTA: MVI  C,0             ;ELSE B$ NOT IN A$
1404 C3D913                JMP  INST5           ;RETURN
```

```
ADDR OBJECT   LINE TAG     OPCD OPRNDS            COMMENTS

1407          3373 EXPR:   EQU  $
              3374 ;
              3375 ;
              3376 ;  EVALUATE EXPRESSION ROUTINE
              3377 ;  LEAVE RESULT IN FACC
              3378 ;  RETURN WHEN EXPRESSION ENDS (TYPICALLY AT END OF LINE)
              3379 ;
              3380 ;
1407 AF       3381         XRA  A                 ;CLEAR REG A
1408 325B21   3382         STA  PARCT             ;SET PAREN CTR
140B EB       3383         XCHG                   ;SAVE H,L
140C 210000   3384         LXI  H,0               ;GET A ZERO
140F 225921   3385         SHLD SPCTR             ;INIT CTR
1412 2AFA21   3386         LHLD PROGE             ;POINT END OF PROGRAM AREA
1415 23       3387         INX  H                 ;POINT ONE MORE
1416 3600     3388         MVI  M,0               ;INIT START OF STACK
1418 224620   3389         SHLD EXPRS             ;SAVE IT
141B EB       3390         XCHG                   ;RESTORE H,L
              3391 ;
141C          3392 LOOKD:  EQU  $                 ;LOOK FOR CONSTANT, VARIABLE, OR
141C CF       3393         RST  1                 ;SKIP TO NON-BLANK
141D CD351B   3394         CALL NUMER             ;GO TEST IF NUMERIC
1420 C23614   3395         JNZ  LDALP             ;BRIF NOT
1423 CD1B0B   3396 LDNUM:  CALL FIN               ;GO CONVERT NUMERIC (PUT TO FACC
1426 44       3397 LDF:    MOV  B,H               ;COPY H,L TO B,C
1427 4D       3398         MOV  C,L               ;SAME
1428 2A4620   3399         LHLD EXPRS             ;GET ADDR OF EXPR AREA
142B CD0B1B   3400         CALL GTEMP             ;GO STORE THE FACC IN TEMP AREA
142E 224620   3401         SHLD EXPRS             ;SAVE UPDATED ADDRESS
1431 60       3402         MOV  H,B               ;RESTORE H
1432 69       3403         MOV  L,C               ;RESTORE L
1433 C38215   3404         JMP  LOOKO             ;GO GET AN OPERATION CODE
1436 FE2E     3405 LDALP:  CPI  '.'               ;SEE IF LEADING DECIMAL POINT
1438 CA2314   3406         JZ   LDNUM             ;BRIF IS
143B CD2C1B   3407         CALL ALPHA             ;GO SEE IF ALPHA
143E C22615   3408         JNZ  LDDTN             ;BRIF NOT
1441 46       3409         MOV  B,M               ;SAVE 1ST CHAR
1442 23       3410         INX  H                 ;POINT NEXT
1443 0E20     3411         MVI  C,' '             ;DEFAULT FOR 1 CHAR VAR
1445 CD351B   3412         CALL NUMER             ;GO SEE IF 2ND IS NUMERIC
1448 C27B14   3413         JNZ  LDFN              ;BRIF NOT
144B 23       3414         INX  H                 ;POINT NEXT
144C 4F       3415         MOV  C,A               ;SAVE THE CHAR
144D CF       3416 LDV1:   RST  1                 ;GET NEXT CHAR
144E FE24     3417         CPI  '$'               ;TEST IF STRING
1450 F5       3418         PUSH PSW               ;SAVE STATUS
1451 C25A14   3419         JNZ  LDV2              ;BRIF NOT
1454 79       3420         MOV  A,C               ;GET LOW CHAR
1455 F680     3421         ORI  128               ;SET STRING
1457 4F       3422         MOV  C,A               ;SAVE IT
1458 23       3423         INX  H                 ;SKIP $
```

```
ADR OBJECT   TAG      OPCD OPRNDS                COMMENTS

1459 CF                RST  1                    ;SKIP SPACES
145A FE28      LDV2:   CPI  '('                  ;TEST IF PAREN
145C CA7E18            JZ   LDV2A                ;BRIF IS
145F E5        LDV2B:  PUSH H                    ;SAVE H,L
1460 50                MOV  D,B                  ;COPY B,C
1461 59                MOV  E,C                  ;TO D,E
1462 CD3F1B            CALL SEARC                ;GO GET VAR ADDR IN D,E
1465 2A4620    LDV:    LHLD EXPRS                ;GET EXPR ADDR
1468 CD241B            CALL SADR                 ;GO STORE ADDRESS
146B 224620            SHLD EXPRS                ;SAVE ADDRESS
146E EB                XCHG                      ;H,L TO D,E
146F E1                POP  H                    ;GET OLD H,L
1470 F1                POP  PSW                  ;GET STATUS
1471 C2B215            JNZ  LOOKO                ;BRIF NOT STRING
1474 EB                XCHG                      ;GET OLD H,L
1475 36E7              MVI  M,231                ;MARK AS STRING ADDRESS
1477 EB                XCHG                      ;RESTORE H,L
1478 C3B215            JMP  LOOKO                ;GO LOOK FOR OPCODE
147B CD2C1B    LDFN:   CALL ALPHA                ;GO SEE IF FUNCTION
147E C24D14            JNZ  LDV1                 ;BRIF IT'S NOT
1481 2B        LDFN1:  DCX  H                    ;POINT BACK TO 1ST
1482 7E                MOV  A,M                  ;GET THAT CHAR
1483 FE20              CPI  ' '                  ;TEST IF SPACE
1485 CA8114            JZ   LDFN1                ;LOOP IF TRUE
1488 E5                PUSH H                    ;SAVE H,L
1489 118E1D            LXI  D,RNDLI              ;POINT LITERAL
148C D7                RST  2                    ;GO COMPARE
148D CAFC14            JZ   LDRND                ;BRIF RND
1490 E1                POP  H                    ;GET H,L
1491 E5                PUSH H                    ;RESAVE
1492 11BE1F            LXI  D,FNLIT              ;POINT LITERAL
1495 D7                RST  2                    ;GO SEE IF FN X
1496 CAD714            JZ   FNL                  ;BRIF IS
1499 E1                POP  H                    ;GET H,L
149A E5                PUSH H                    ;RESAVE
149B 113B1E            LXI  D,PILIT              ;POINT LIT
149E D7                RST  2                    ;GO COMPARE
149F CA0A15            JZ   LDPI                 ;BRIF PI
14A2 E1        FUNC0:  POP  H                    ;GET H,L
14A3 11721D            LXI  D,FUNCT              ;POINT FUNCTION TABLE
14A6 E5        FUNC1:  PUSH H                    ;SAVE POINTER
14A7 1A                LDAX D                    ;GET 1ST BYTE
14A8 B7                ORA  A                    ;TEST IF END OF TABLE
14A9 CACF14            JZ   FUNC4                ;BRIF IS
14AC D7                RST  2                    ;GO COMPARE
14AD CABB14            JZ   FUNC3                ;BRIF EQUAL
14B0 CD041B    FUNC2:  CALL SKP2Z                ;FIND END LIT
14B3 13                INX  D                    ;POINT NEXT
14B4 13                INX  D                    ;POINT NEXT
14B5 13                INX  D                    ;SAME
14B6 13                INX  D                    ;AGAIN
```

```
ADDR OBJECT   LINE TAG     OPCD OPRNDS        COMMENTS

14B7 E1                    POP  H             ;GET ADDR
14B8 C3A614                JMP  FUNC1         ;LOOP
14BB CD041B        FUNC3:  CALL SKP2Z         ;FIND END OF LIT
14BE 13                    INX  D             ;POINT NEXT
14BF 1A                    LDAX D             ;GET A BYTE LOW
14C0 4F                    MOV  C,A           ;SAVE IT
14C1 13                    INX  D             ;POINT NEXT
14C2 1A                    LDAX D             ;GET HI BYTE
14C3 47                    MOV  B,A           ;SAVE IT (B,C = ADDR OF FUNC)
14C4 CF                    RST  1             ;SKIP BLANKS
14C5 FE28                  CPI  '('           ;TEST FOR OPEN PAREN
14C7 C2A01A                JNZ  SNERR         ;BRIF MISSING PAREN
14CA 13                    INX  D             ;POINT TYPE CODE
14CB 1A                    LDAX D             ;LOAD IT
14CC C31415                JMP  LDFNC         ;CONTINUE
14CF E1            FUNC4:  POP  H             ;GET H,L
14D0 46                    MOV  B,M           ;GET 1ST CHAR
14D1 0E20                  MVI  C,' '         ;SPACE 2ND CHAR
14D3 23                    INX  H             ;POINT TO NEXT
14D4 C34D14                JMP  LDV1          ;BRIF VARIABLE
14D7 D1            FNL:    POP  D             ;DUMMY RESET STACK POINTER
14D8 CD041B                CALL VAR           ;GO GET FN NAME
14DB 42                    MOV  B,D           ;COPY TO B,C
14DC 4B                    MOV  C,E           ;SAME
14DD EB            FNL2:   XCHG               ;SAVE H,L
14DE 2A4620                LHLD EXPRS         ;POINT EXPR STACK
14E1 23                    INX  H             ;POINT NEXT
14E2 70                    MOV  M,B           ;MOVE THE LETTER
14E3 23                    INX  H             ;POINT NEXT
14E4 71                    MOV  M,C           ;MOVE DIGIT ($??)
14E5 23                    INX  H             ;POINT NEXT
14E6 36AF                  MVI  M,175         ;MOVE CODE
14E8 79                    MOV  A,C           ;GET LO NAME
14E9 B7                    ORA  A             ;TEST IT
14EA F2EF14                JP   FNL3          ;BRIF NOT STRING
14ED 36CF                  MVI  M,207         ;MOVE CODE
14EF 224620        FNL3:   SHLD EXPRS         ;SAVE POINTER
14F2 EB                    XCHG               ;GET H,L
14F3 CF                    RST  1             ;GET NEXT CHAR
14F4 FE28                  CPI  '('           ;TEST IF OPEN PAREN
14F6 C2A01A                JNZ  SNERR         ;BRIF NOT
14F9 C31C14                JMP  LOOKD         ;CONTINUE
14FC FE28          LDRND:  CPI  '('           ;TEST IF RND(X)
14FE CAA214                JZ   FUNC0         ;BRIF IS
1501 E5                    PUSH H             ;ELSE, SAVE H,L
1502 CDE411                CALL RND           ;GO GET RANDOM NUMBER
1505 E1                    POP  H             ;RESTORE H,L
1506 D1                    POP  D             ;RESTORE STACK POINTER
1507 C32614                JMP  LDF           ;ACT AS IF CONSTANT
150A 3C            LDPI:   INR  A             ;SET NON ZERO
150B D1                    POP  D             ;DUMMY STACK POP
```

```
ADDR OBJECT  TAG     OPCD OPRNDS      COMMENTS

150C F5              PUSH PSW         ;SAVE STATUS
150D E5              PUSH H           ;SAVE H,L
150E 114A1E          LXI  D,PI        ;GET ADDRESS OF 3.1415
1511 C36514          JMP  LDV         ;GO ACT LIKE VARIABLE
1514 D1      LDFNC:  POP  D           ;POP THE STACK
1515 EB              XCHG             ;FLIP/FLOP
1516 2A4620          LHLD EXPRS       ;GET ADDR
1519 23              INX  H           ;POINT NEXT
151A 70              MOV  M,B         ;HIGH ADDR
151B 23              INX  H           ;POINT NEXT
151C 71              MOV  M,C         ;LOW ADDR
151D 23              INX  H           ;POINT NEXT
151E 77              MOV  M,A         ;CODE
151F 224620          SHLD EXPRS       ;SAVE ADDR
1522 EB              XCHG             ;RESTORE H,L
1523 C31C14          JMP  LOOKD       ;NEXT MUST BE DATA TOO
1526 FE2D    LDDTN:  CPI  '-'         ;TEST IF UNARY MINUS
1528 C23A15          JNZ  LDDTP       ;BRIF NOT
152B EB              XCHG             ;SAVE H,L
152C 2A4620          LHLD EXPRS       ;GET EXPR END
152F 23              INX  H           ;POINT ONE MORE
1530 3661            MVI  M,97        ;CODE FOR NEG
1532 224620          SHLD EXPRS       ;RESTORE PTR
1535 EB              XCHG             ;RESTORE H,L
1536 23      SKPP:   INX  H           ;POINT PAST THIS BYTE
1537 C31C14          JMP  LOOKD       ;NEXT MUST BE DATA
153A FE2B    LDDTP:  CPI  '+'         ;TEST IF UNARY PLUS
153C CA3615          JZ   SKPP        ;IGNORE IF IS
153F FE28            CPI  '('         ;ELSE, TEST IF OPEN PAREN
1541 CAA015          JZ   OBRCE       ;BRIF IS
1544 FE27            CPI  39          ;TEST IF LITERAL (SINGLE QUOTE)
1546 CA4E15          JZ   LITST       ;BRIF IS
1549 FE22            CPI  '"'         ;TEST IF LITERAL
154B C2A01A          JNZ  SNERR       ;BRIF NOT CONSTANT, FUNCTION, OR
154E 4F      LITST:  MOV  C,A         ;SAVE DELIMITER
154F 115820          LXI  D,STRIN     ;POINT BUFFER
1552 06FF            MVI  B,255       ;INIT CTR
1554 23      LIT1:   INX  H           ;POINT NEXT
1555 7E              MOV  A,M         ;LOAD NEXT
1556 13              INX  D           ;POINT NEXT
1557 12              STAX D           ;STORE IT
1558 B7              ORA  A           ;TEST IF END
1559 CAA01A          JZ   SNERR       ;BRIF ERROR
155C 04              INR  B           ;COUNT IT
155D B9              CMP  C           ;TEST IF END OF STRING
155E C25415          JNZ  LIT1        ;BRIF NOT
1561 23              INX  H           ;POINT NEXT
1562 115820          LXI  D,STRIN     ;POINT BEGIN
1565 78              MOV  A,B         ;GET COUNT
1566 12              STAX D           ;PUT COUNT
1567 1F              RAR              ;DIVIDE BY TWO
```

```
ADDR OBJECT   TAG     OPCD OPRNDS              COMMENTS

1568 3C               INR  A                   ;PLUS ONE
1569 4F               MOV  C,A                 ;SAVE IT
156A 0600             MVI  B,0                 ;ZERO HIGH
156C E5               PUSH H                   ;SAVE PTR
156D 2A5921           LHLD SPCTR               ;GET CTR
1570 09               DAD  B                   ;PLUS OLD
1571 225921           SHLD SPCTR               ;UPDATE IT
1574 D1               POP  D                   ;GET OLD H,L
1575 210000           LXI  H,0                 ;GET A ZERO
1578 E5       LIT2:   PUSH H                   ;GET 2 WORK BYTES
1579 0D               DCR  C                   ;SUB 1 FROM COUNT
157A C27815           JNZ  LIT2                ;CONTINUE
157D 39               DAD  SP                  ;GET ADDR OF STACK
157E D5               PUSH D                   ;SAVE PTR TO STMT
157F EB               XCHG                     ;SAVE H,L IN D,E
1580 2A4620           LHLD EXPRS               ;GET START OF EXPR
1583 23               INX  H                   ;PLUS ONE
1584 72               MOV  M,D                 ;HI BYTE
1585 23               INX  H                   ;POINT NEXT
1586 73               MOV  M,E                 ;LO BYTE
1587 23               INX  H                   ;POINT NEXT
1588 36E7             MVI  M,231               ;TYPE CODE
158A 224620           SHLD EXPRS               ;SAVE ADDR
158D EB               XCHG                     ;D,E BACK TO H,L
158E 115820           LXI  D,STRIN             ;POINT STRING AREA
1591 1A               LDAX D                   ;GET COUNT
1592 3C               INR  A                   ;ADD ONE TO COUNT
1593 47               MOV  B,A                 ;SAVE CTR
1594 1A       LIT3:   LDAX D                   ;GET A BYTE
1595 77               MOV  M,A                 ;STORE IT
1596 23               INX  H                   ;POINT NEXT
1597 13               INX  D                   ;DITTO
1598 05               DCR  B                   ;DECR CTR
1599 C29415           JNZ  LIT3                ;LOOP
159C E1       LIT4:   POP  H                   ;RESTORE H,L
159D C3B215           JMP  LOOKO               ;NEXT IS OP
15A0 EB       OBRCE:  XCHG                     ;SAVE H,L
15A1 215871           LXI  H,PARCT             ;POINT PAREN COUNT
15A4 34               INR  M                   ;ADD 1
15A5 2A4620           LHLD EXPRS               ;GET ADDR
15A8 23               INX  H                   ;POINT NEXT
15A9 3605             MVI  M,5                 ;PUT CODE
15AB 224620           SHLD EXPRS               ;SAVE ADDR
15AE EB               XCHG                     ;RESTORE H,L
15AF C33615           JMP  SKPP                ;GO SKIP CHAR
15B2 CF       LOOKO:  RST  1                   ;SKIP BLANKS
15B3 FE2B             CPI  '+'                 ;TEST IF PLUS
15B5 0621             MVI  B,33                ;CODE
15B7 CAED15           JZ   OP1                 ;BRIF IS
15BA FE2D             CPI  '-'                 ;TEST IF MINUS
15BC 0625             MVI  B,37
```

```
ADDR OBJECT   LINE TAG     OPCD OPRNDS                 COMMENTS

15BE CAED15                JZ   OP1                    ;BRIF IS
15C1 FE2F                  CPI  '/'                    ;TEST IF DIVIDE
15C3 0645                  MVI  B,69                   ;CODE
15C5 CAED15                JZ   OP1                    ;BRIF IS
15C8 FE5E                  CPI  '^'                    ;TEST IF EXPON
15CA 0681                  MVI  B,129                  ;CODE
15CC CAED15                JZ   OP1                    ;BRIF IS
15CF FE29                  CPI  ')'                    ;TEST IF CLOSE PAREN
15D1 CA5016                JZ   OP3                    ;BRIF IS
15D4 FE2C                  CPI  ','                    ;TEST IF COMMA
15D6 CA3B16                JZ   OP2                    ;BRIF IS
15D9 FE2A                  CPI  '*'                    ;TEST IF MULTIPLY
15DB 0641                  MVI  B,65                   ;CODE
15DD CAED15                JZ   OP1                    ;BRIF IS
                   ; ELSE MUST BE END OF EXPRESSION
15E0 3A5821        ENDXP:  LDA  PARCT                  ;GET OPEN PAREN COUNT
15E3 B7                    ORA  A                      ;TEST IT
15E4 C2A01A                JNZ  SNERR                  ;BRIF # OF ('S NOT = # OF )'S
15E7 221620                SHLD ADDR3                  ;SAVE ADDR OF STMT
15EA C35E16                JMP  EVAL                   ;GO EVALUATE
15ED E5            OP1:    PUSH H                      ;SAVE H,L
15EE 110100                LXI  D,1                    ;INIT BYTE COUNTER
15F1 2A4620                LHLD EXPRS                  ;GET END OF EXPR
15F4 78                    MOV  A,B                    ;GET OP
15F5 E6E3                  ANI  227                    ;TURN OFF ALL BUT PRTY BITS
15F7 4F                    MOV  C,A                    ;SAVE IT
15F8 7E            OPLP1:  MOV  A,M                    ;GET OP
15F9 E6E3                  ANI  227                    ;TURN OFF ALL BUT PRTY BITS
15FB B9                    CMP  C                      ;COMPARE PREV TO THIS
15FC DA1816                JC   INS1                   ;BRIF OP LESS THAN THIS
15FF CA0D16                JZ   INS0                   ;OR EQUAL
1602 E603          OPLP3:  ANI  3                      ;ISOLATE LENGTH
1604 13            OPLP2:  INX  D                      ;ADD CTR
1605 2B                    DCX  H                      ;POINT PRIOR
1606 3D                    DCR  A                      ;SUB 1 FROM COUNT
1607 C20416                JNZ  OPLP2                  ;LOOP
160A C3F815                JMP  OPLP1                  ;LOOP OUTER
160D E620          INS0:   ANI  32                     ;TEST IF PLUS/MINUS
160F CA1716                JZ   INS3                   ;BRIF NOT
1612 2B                    DCX  H                      ;ELSE, POINT PRIOR
1613 13                    INX  D                      ;ADD TO COUNT
1614 C32A16                JMP  INS                    ;BR AROUND
1617 7E            INS3:   MOV  A,M                    ;RELOAD OP
1618 FE61          INS1:   CPI  97                     ;TEST IF UNARY MINUS (TREAT AS V
161A CA2A16                JZ   INS                    ;BRIF IS
161D E6E0                  ANI  224                    ;MASK
161F CA2A16                JZ   INS                    ;BRIF AT COMMA PAREN OR START OF
1622 110400                LXI  D,4                    ;INIT COUNT
1625 2A4620                LHLD EXPRS                  ;POINT START
1628 FF                    RST  7                      ;ADJUST TO FIRST OP
1629 FD                    DB   -3
```

```
ADDR OBJECT    LINE TAG    OPCD OPRNDS          COMMENTS

162A 23        3679 INS:   INX  H               ;POINT NEXT
162B 4E        3680        MOV  C,M             ;SAVE CHAR AT (H,L)
162C 70        3681        MOV  M,B             ;ROTATE TO (H,L)
162D 41        3682        MOV  B,C             ;COPY
162E 1B        3683        DCX  D               ;DECR CTR
162F 7A        3684        MOV  A,D             ;GET HI
1630 B3        3685        ORA  E               ;OR WITH LO
1631 C22A16    3686        JNZ  INS             ;LOOP
1634 224620    3687        SHLD EXPRS           ;SAVE NEW BEGIN
1637 E1        3688        POP  H               ;RESTORE H,L
1638 C33615    3689        JMP  SKPP            ;NEXT AFTER THIS IS DATA
163B 3A5821    3690 OP2:   LDA  PARCT           ;GET OPEN PAREN COUNT
163E B7        3691        ORA  A               ;TEST IT
163F CAE015    3692        JZ   ENDXP           ;BRIF END OF EXPR
1642 EB        3693        XCHG                 ;ELSE SAVE H,L
1643 2A4620    3694        LHLD EXPRS           ;GET EXPR BEGIN
1646 23        3695        INX  H               ;POINT NEXT
1647 3601      3696        MVI  M,1             ;MOVE A COMMA
1649 224620    3697        SHLD EXPRS           ;UPDATE POINTER
164C EB        3698        XCHG                 ;FLIP BACK
164D C33615    3699        JMP  SKPP
1650 3A5821    3700 OP3:   LDA  PARCT           ;GET OPEN PAREN COUNT
1653 3D        3701        DCR  A               ;SUBTRACT ONE
1654 325821    3702        STA  PARCT           ;SAVE IT
1657 FAA01A    3703        JM   SNERR           ;BRIF TOO MANY )'S
165A 23        3704        INX  H               ;POINT NEXT SOURCE
165B 221620    3705        SHLD ADDR3           ;SAVE ADDR
165E 2A4620    3706 EVAL:  LHLD EXPRS           ;GET END OF EXPR
1661 010000    3707 EV0:   LXI  B,0             ;INIT B,C TO ZERO
1664 04        3708 EV1:   INR  B               ;COUNT EACH BYTE
1665 7E        3709        MOV  A,M             ;GET CODE IN REG A
1666 2B        3710        DCX  H               ;POINT NEXT
1667 FEE3      3711        CPI  227             ;TEST IF DATA
1669 C27416    3712        JNZ  EV2             ;BRIF NOT DATA
166C 2B        3713 EV1A:  DCX  H               ;POINT NEXT
166D 2B        3714        DCX  H               ;DITTO
166E 04        3715        INR  B               ;BUMP CTR
166F 04        3716        INR  B               ;BY TWO
1670 0C        3717        INR  C               ;COUNT THE TERM
1671 C36416    3718        JMP  EV1             ;LOOP
1674 FEAF      3719 EV2:   CPI  175             ;TEST IF NUMERIC USER FN
1676 CA781C    3720        JZ   FN              ;BRIF IS
1679 FECF      3721        CPI  207             ;TEST IF STRING USER FN
167B CA781C    3722        JZ   FN              ;BRIF IS
167E F5        3723        PUSH PSW             ;ELSE, SAVE STATUS
167F E6E3      3724        ANI  227             ;MASK IT
1681 FEA3      3725        CPI  163             ;TEST IF NUMERIC FUNCTION
1683 CA9416    3726        JZ   EV2A            ;BRIF IS
1686 FEC3      3727        CPI  195             ;TEST IF STRING FUNCTION
1688 CA9416    3728        JZ   EV2A            ;BRIF IS
168B F1        3729        POP  PSW             ;RESTORE CODE
```

```
ADDR OBJECT   LINE TAG    OPCD OPRNDS     COMMENTS

168C FEE7     3730        CPI  231        ;TEST IF STRING ADDR
168E CA6C16   3731        JZ   EV1A       ;BRIF IS
1691 C31C17   3732        JMP  EV5        ;BR AROUND
1694 23       3733 EV2A:  INX  H          ;RESET TO TYPE CODE
1695 221220   3734        SHLD ADDR1      ;SAVE ADDR
1698 D1       3735        POP  D          ;DUMMY POP
1699 C5       3736        PUSH B          ;SAVE CTRS
169A 2B       3737        DCX  H          ;POINT TO LOW JMP ADDR
169B 5E       3738        MOV  E,M        ;LOW BYTE
169C 2B       3739        DCX  H          ;POINT BACK
169D 56       3740        MOV  D,M        ;HIGH BYTE
169E 221420   3741        SHLD ADDR2      ;SAVE LOCATION
16A1 21AB16   3742        LXI  H,EV3      ;GET RETURN ADDRESS
16A4 E5       3743        PUSH H          ;SAVE ON STACK
16A5 D5       3744        PUSH D          ;SAVE ADDRESS
16A6 CD631C   3745        CALL ARG        ;GO GET 1ST ARG
16A9 E1       3746        POP  H          ;GET H,L ADDRESS
16AA E9       3747        PCHL            ;GO EXECUTE THE FUNCTION
16AB          3748 EV3:   EQU  $          ;FUNCTIONS RETURN HERE
16AB 2A1420   3749        LHLD ADDR2      ;GET ADDR FUNC
16AE 23       3750        INX  H          ;POINT LO
16AF 23       3751        INX  H          ;POINT TYPE
16B0 7E       3752        MOV  A,M        ;LOAD IT
16B1 E6E0     3753        ANI  224        ;MASK IT
16B3 FEC0     3754        CPI  192        ;TEST IF STRING
16B5 CAF016   3755        JZ   EV4        ;BRIF IS
16B8 C1       3756        POP  B          ;GET CTRS
16B9 2A5921   3757        LHLD SPCTR      ;GET COUNTER
16BC 23       3758        INX  H          ;PLUS
16BD 23       3759        INX  H          ;TWO WORDS
16BE 225921   3760        SHLD SPCTR      ;STORE IT
16C1 210000   3761        LXI  H,0        ;LOAD ZERO TO H,L
16C4 E5       3762        PUSH H          ;GET BLOCK OF
16C5 E5       3763        PUSH H          ;BYTES
16C6 39       3764        DAD  SP         ;GET STACK ADDR
16C7 C5       3765        PUSH B          ;SAVE CTRS
16C8 E5       3766        PUSH H          ;SAVE ADDR
16C9 F7       3767        RST  6          ;GO STORE THE VARIABLE
16CA 3EE3     3768        MVI  A,227      ;TYPE=NUM
16CC D1       3769 EV3A:  POP  D          ;GET ADDR IN STACK
16CD 2A1220   3770        LHLD ADDR1      ;GET ADDR 1ST ARG
16D0 77       3771        MOV  M,A        ;STORE TYPE CODE
16D1 2B       3772        DCX  H          ;POINT ONE BACK
16D2 73       3773        MOV  M,E        ;STORE LO ADDR
16D3 2B       3774        DCX  H          ;POINT BACK
16D4 72       3775        MOV  M,D        ;STORE HI ADDR
16D5 2A1420   3776        LHLD ADDR2      ;GET LOCATION FUNCTION
16D8 23       3777        INX  H          ;POINT LO
16D9 23       3778        INX  H          ;POINT TYPE
16DA 7E       3779        MOV  A,M        ;LOAD TYPE
16DB 46       3780        MOV  B,M        ;GET TYPE
```

```
ADDR OBJECT   LINE TAG     OPCD OPRNDS         COMMENTS

16DC FF       3781          RST  7             ;ADJUST H,L
16DD FD       3782          DB   -3
16DE 78       3783          MOV  A,B           ;LOAD TYPE
16DF C1       3784          POP  B             ;RESTORE CTRS
16E0 E618     3785          ANI  24            ;ISOLATE #ARGS
16E2 1F       3786          RAR                ;SHIFT RIGHT
16E3 1F       3787          RAR                ;AGAIN
16E4 1F       3788          RAR                ;ONCE MORE
16E5 57       3789          MOV  D,A           ;SAVE IT
16E6 82       3790          ADD  D             ;TIMES 2
16E7 82       3791          ADD  D             ;TIMES 3
16E8 04       3792          INR  B             ;POINT
16E9 04       3793          INR  B             ;1ST POSIT IN LOC
16EA CDED1A   3794          CALL SQUIS         ;GO COMPRESS STACK
16ED C35E16   3795          JMP  EVAL          ;START AT BEGINNING
16F0 115820   3796 EV4:     LXI  D,STRIN       ;POINT STRING BUFFER
16F3 1A       3797          LDAX D             ;LOAD IT
16F4 1F       3798          RAR                ;DIVIDE BY TWO
16F5 3C       3799          INR  A             ;ADD 1
16F6 2A5921   3800          LHLD SPCTR         ;GET SP COUNT
16F9 4F       3801          MOV  C,A           ;SAVE LO
16FA 0600     3802          MVI  B,0           ;SET HI
16FC 09       3803          DAD  B             ;ADD NUMBER WORDS
16FD 225921   3804          SHLD SPCTR         ;SAVE SP COUNT
1700 210000   3805          LXI  H,0           ;GET SOME ZEROS
1703 C1       3806          POP  B             ;GET CTRS
1704 E5       3807 EV4A:    PUSH H             ;GET 1 WORD
1705 3D       3808          DCR  A             ;DECR CTR
1706 C20417   3809          JNZ  EV4A          ;LOOP
1709 39       3810          DAD  SP            ;GET ADDRESS IN H,L
170A C5       3811          PUSH B             ;RE-SAVE CTRS
170B E5       3812          PUSH H             ;SAVE ADDR
170C 1A       3813          LDAX D             ;GET COUNT
170D 3C       3814          INR  A             ;PLUS ONE
170E 47       3815          MOV  B,A           ;SAVE IT
170F 1A       3816 EV4B:    LDAX D             ;GET A BYTE
1710 77       3817          MOV  M,A           ;STORE IT
1711 13       3818          INX  D             ;POINT NEXT
1712 23       3819          INX  H             ;DITTO
1713 05       3820          DCR  B             ;DECR CTR
1714 C20F17   3821          JNZ  EV4B          ;LOOP
1717 3EE7     3822          MVI  A,231         ;TYPE CODE
1719 C3CC16   3823          JMP  EV3A          ;CONTINUE
171C FE05     3824 EV5:     CPI  5             ;TEST IF OPEN PAREN
171E C23A17   3825          JNZ  EV6           ;BRIF NOT
1721 3E01     3826          MVI  A,1           ;DELETE 1 BYTE
1723 CDED1A   3827          CALL SQUIS         ;GO COMPRESS IT
1726 2A1620   3828          LHLD ADDR3         ;RESTORE STMT POINTER
1729 3A6821   3829          LDA  DIMSW         ;GET SUBSR SWITCH
172C B7       3830          ORA  A             ;TEST IT
172D CAB215   3831          JZ   LOOKO         ;BRIF NOT IN SUBSCRIPT
```

```
ADDR OBJECT   TAG    OPCD OPRNDS      COMMENTS

1730 3A5821          LDA  PARCT       ;GET OPEN PAREN COUNT
1733 B7              ORA  A           ;TEST
1734 C2B215          JNZ  LOOKO       ;BRIF NOT ZERO
1737 C35E16          JMP  EVAL        ;ELSE EVALUATE COMPLETE SUBSCR
173A B7       EV6:   ORA  A           ;TEST IF END OF EXPRESSION
173B C26B17          JNZ  EV9         ;BRIF NOT
173E 3A6821          LDA  DIMSW       ;GET DIM SW
1741 B7              ORA  A           ;TEST IT
1742 C44018          CNZ  EDM1        ;BRIF NOT OFF
1745 79              MOV  A,C         ;GET TERM COUNT
1746 FE01            CPI  1           ;TEST IF ONE
1748 C29C1A          JNZ  STERR       ;ERROR IF NOT ONE
174B 23              INX  H           ;POINT HIGH ADDR
174C 23              INX  H           ;SAME
174D 56              MOV  D,M         ;HIGH TO D
174E 23              INX  H           ;POINT LOW
174F 5E              MOV  E,M         ;LOW TO E
1750 CD2618          CALL EVLD        ;GO LOAD VALUE
1753 2A5921          LHLD SPCTR       ;GET STACK CTR
1756 7D       EV7:   MOV  A,L         ;GET LO BYTE
1757 B4              ORA  H           ;PLUS HI
1758 CA6017          JZ   EV8         ;BRIF ZERO
175B D1              POP  D           ;RETURN 2 BYTES
175C 2B              DCX  H           ;DECR CTR
175D C35617          JMP  EV7         ;LOOP
1760 3A6821   EV8:   LDA  DIMSW       ;GET DIM SW
1763 B7              ORA  A           ;TEST IT
1764 C46719          CNZ  EDM4        ;BRIF ON
1767 2A1620          LHLD ADDR3       ;RESTORE STMT PTR
176A C9              RET              ;RETURN TO STMT PROCESSOR
176B FE21     EV9:   CPI  33          ;TEST IF PLUS
176D 11BF17          LXI  D,FADDJ     ;ADDR
1770 CA9D17          JZ   EV10        ;BRIF IS
1773 FE25            CPI  37          ;TEST IF MINUS
1775 11530E          LXI  D,FSUB      ;ADDR
1778 CA9D17          JZ   EV10        ;BRIF IS
177B FE41            CPI  65          ;TEST IF MUL
177D 116C0E          LXI  D,FMUL      ;ADDR
1780 CA9D17          JZ   EV10        ;BRIF IS
1783 FE45            CPI  69          ;TEST IF DIV
1785 11E30E          LXI  D,FDIV      ;ADDR
1788 CA9D17          JZ   EV10        ;BRIF IS
178B FE01            CPI  1           ;TEST IF COMMA
178D CA1A18          JZ   EVCOM       ;BRIF IS
1790 FE61            CPI  97          ;TEST IF UNARY MINUS
1792 CA0618          JZ   EVNEG       ;BRIF IS
1795 FE81            CPI  129         ;TEST IF EXPONENTIAL
1797 11C717          LXI  D,POWER     ;ADDR
179A C29C1A          JNZ  STERR       ;ERROR IF NOT
179D 23       EV10:  INX  H           ;POINT TO
179E 23              INX  H           ;1ST DATA
```

```
ADDR OBJECT     LINE TAG      OPCD OPRNDS             COMMENTS

179F C5         3883          PUSH B                  ;SAVE CTRS
17A0 D5         3884          PUSH D                  ;SAVE ROUTINE ADDR
17A1 56         3885          MOV  D,M                ;HIGH TO D
17A2 23         3886          INX  H                  ;POINT NEXT
17A3 5E         3887          MOV  E,M                ;LOW TO E
17A4 E5         3888          PUSH H                  ;SAVE POINTER
17A5 CD2618     3889          CALL EVLD               ;GO LOAD VALUE
17A8 E1         3890          POP  H                  ;RESTORE H,L
17A9 23         3891          INX  H                  ;POINT 2ND DATA
17AA 23         3892          INX  H                  ;SAME
17AB 56         3893          MOV  D,M                ;HIGH TO D
17AC 23         3894          INX  H                  ;POINT NEXT
17AD 5E         3895          MOV  E,M                ;LOW TO E
17AE 23         3896          INX  H                  ;POINT NEXT
17AF 3AF521     3897          LDA  NS                 ;GET PREV TYPE
17B2 BE         3898          CMP  M                  ;TEST THIS TYPE
17B3 C2A01A     3899          JNZ  SNERR              ;BRIF MIXED MODE
17B6 2B         3900          DCX  H                  ;POINT BACK
17B7 E3         3901          XTHL                    ;POP ADDR FROM STACK, PUSH H ONT
17B8 01F517     3902          LXI  B,EV11             ;RETURN ADDRESS
17BB C5         3903          PUSH B                  ;SAVE ON STACK
17BC E5         3904          PUSH H                  ;SAVE JUMP ADDR
17BD EB         3905          XCHG                    ;PUT VAR ADDR TO H,L
17BE C9         3906          RET                     ;FAKE CALL TO ROUTINE
17BF FEE7       3907 FADDJ:   CPI  231                ;TEST IF STRINGS
17C1 CA7C12     3908          JZ   CONCA              ;BRIF IS
17C4 C3590D     3909          JMP  FADD               ;ELSE, GO ADD
17C7 E5         3910 POWER:   PUSH H                  ;SAVE ADDR OF VAR
17C8 212420     3911          LXI  H,TEMP1            ;POINT SAVE AREA
17CB F7         3912          RST  6                  ;SAVE X
17CC E1         3913          POP  H                  ;RESTORE H,L
17CD EF         3914          RST  5                  ;LOAD IT
17CE 213C20     3915          LXI  H,TEMP7            ;POINT SAVE AREA
17D1 F7         3916          RST  6                  ;SAVE B
17D2 21421E     3917          LXI  H,TWO              ;POINT CONSTANT: 2
17D5 CD530E     3918          CALL FSUB               ;GO SEE IF X^2
17D8 CAEB17     3919          JZ   XSQR               ;BRIF X^2
17DB 212420     3920          LXI  H,TEMP1            ;POINT X
17DE EF         3921          RST  5                  ;GO LOAD IT
17DF CDA510     3922          CALL LOG                ;GET NATURAL LOGRITHM
17E2 213C20     3923          LXI  H,TEMP7            ;POINT B
17E5 CD6C0E     3924          CALL FMUL               ;GO MULTIPLY
17E8 C3EC10     3925          JMP  EXP                ;GET EXP FUNC
                3926 ; X^B = EXP(B*LOG(X))
17EB 212420     3927 XSQR:    LXI  H,TEMP1            ;POINT X
17EE EF         3928          RST  5                  ;LOAD X
17EF 212420     3929          LXI  H,TEMP1            ;POINT X
17F2 C36C0E     3930          JMP  FMUL               ;TIMES X
17F5 E1         3931 EV11:    POP  H                  ;GET H,L
17F6 C1         3932          POP  B                  ;GET CTRS
17F7 2B         3933          DCX  H                  ;POINT BACK
```

```
ADDR OBJECT   TAG     OPCD OPRNDS         COMMENTS

17F8 2B               DCX  H              ;AND AGAIN
17F9 CD0B1B           CALL GTEMP          ;GO SAVE FACC
17FC FF               RST  7              ;ADJUST H,L
17FD F9               DB   -7
17FE 3E04             MVI  A,4            ;DELETE 4 BYTES
1800 CDED1A           CALL SQUIS          ;GO COMPRESS
1803 C35E16           JMP  EVAL           ;CONTINUE
1806 23       EVNEG:  INX  H              ;POINT BACK TO OP
1807 C5               PUSH B              ;SAVE CTRS
1808 E5               PUSH H              ;SAVE H,L
1809 23               INX  H              ;DITTO
180A 56               MOV  D,M            ;GET HI BYTE
180B 23               INX  H              ;POINT NEXT
180C 5E               MOV  E,M            ;GET LO BYTE
180D CD2618           CALL EVLD           ;GO LOAD VAR
1810 CDDA11           CALL NEG            ;GO NEGATE IT
1813 E1               POP  H              ;GET LOCATION
1814 C1               POP  B              ;GET CTRS
1815 CD0B1B           CALL GTEMP          ;GO STORE FACC IN STACK
1818 FF               RST  7              ;ADJUST H,L
1819 FC               DB   -4
181A 3E01     EVCOM:  MVI  A,1            ;DELETE 1 BYTE
181C CDED1A           CALL SQUIS          ;COMPRESS
181F 215B21           LXI  H,CMACT        ;GET COUNT
1822 34               INR  M              ;INCR
1823 C35E16           JMP  EVAL           ;CONTINUE
1826 23       EVLD:   INX  H              ;POINT TYPE
1827 7E               MOV  A,M            ;LOAD IT
1828 32F521           STA  NS             ;SAVE IT
182B EB               XCHG                ;SAVE H,L IN D,E
182C FEE7             CPI  231            ;TEST IF STRING
182E C22800           JNZ  RST5           ;LOAD FLOATING POINT
1831 115820           LXI  D,STRIN        ;POINT BUFFER
1834 7E               MOV  A,M            ;GET COUNT
1835 3C               INR  A              ;ADD ONE
1836 47               MOV  B,A            ;SAVE COUNT
1837 7E       EVLD1:  MOV  A,M            ;GET NEXT
1838 12               STAX D              ;STORE IT
1839 23               INX  H              ;POINT NEXT
183A 13               INX  D              ;DITTO
183B 05               DCR  B              ;DECR COUNT
183C C23718           JNZ  EVLD1          ;LOOP
183F C9               RET                 ;RETURN
              ;
1840 79       EDM1:   MOV  A,C            ;GET ITEM COUNT
1841 E5               PUSH H              ;SAVE H,L
1842 FE01             CPI  1              ;TEST IF 1
1844 C25318           JNZ  EDM3           ;BRIF NOT
1847 0604             MVI  B,4            ;GET COUNT
1849 212420           LXI  H,TEMP1        ;POINT AREA
184C CD331C           CALL ZEROM          ;GO ZERO IT
```

```
ADDR OBJECT   LINE TAG     OPCD OPRNDS         COMMENTS

184F E1       3985 EDM2A:  POP  H              ;RESTORE H,L
1850 0E01     3986         MVI  C,1            ;SET COUNT
1852 C9       3987         RET                 ;RETURN
1853 FE02     3988 EDM3:   CPI  2              ;TEST IF 2
1855 C2A01A   3989         JNZ  SNERR          ;ELSE, ERROR
1858 FF       3990         RST  7              ;POINT 2ND ARG
1859 05       3991         DB   5
185A 56       3992         MOV  D,M            ;GET HI ADDR
185B 23       3993         INX  H              ;POINT NEXT
185C 5E       3994         MOV  E,M            ;GET LO ADDR
185D CD2618   3995         CALL EVLD           ;LOAD THE ARG
1860 212420   3996         LXI  H,TEMP1        ;POINT AREA
1863 F7       3997         RST  6              ;SAVE THE ARG
1864 C34F18   3998         JMP  EDM2A          ;CONTINUE
1867 CD3B1C   3999 EDM4:   CALL FBIN           ;GET BIN OF FACC
186A 4F       4000         MOV  C,A            ;PUT TO C
186B 0600     4001         MVI  B,0            ;CLEAR HI
186D C5       4002         PUSH B              ;SAVE COL
186E 212420   4003         LXI  H,TEMP1        ;POINT 2ND ARG
1871 EF       4004         RST  5              ;LOAD IT
1872 CD3B1C   4005         CALL FBIN           ;GET BIN IN REG A
1875 5F       4006         MOV  E,A            ;MOVE LO
1876 1600     4007         MVI  D,0            ;CLEAR HI
1878 C1       4008         POP  B              ;GET COL
1879 AF       4009         XRA  A              ;GET A ZERO
187A 326821   4010         STA  DIMSW          ;RESET SW
187D C9       4011         RET                 ;RETURN
187E 78       4012 LDV2A:  MOV  A,B            ;GET HI NAME
187F F680     4013         ORI  128            ;SET BIT
1881 47       4014         MOV  B,A            ;RESTORE
1882 C5       4015         PUSH B              ;SAVE NAME
1883 EB       4016         XCHG                ;SAVE H,L IN D,E
1884 3A5821   4017         LDA  PARCT          ;GET PAREN COUNT
1887 F5       4018         PUSH PSW            ;SAVE
1888 AF       4019         XRA  A              ;CLEAR REG A
1889 325821   4020         STA  PARCT          ;RESET COUNT
188C 2A5921   4021         LHLD SPCTR          ;GET STACK COUNTER
188F E5       4022         PUSH H              ;SAVE IT
1890 210000   4023         LXI  H,0            ;GET A ZERO
1893 225921   4024         SHLD SPCTR          ;RESET CTR
1896 2A4620   4025         LHLD EXPRS          ;GET EXPRST
1899 E5       4026         PUSH H              ;SAVE IT
189A 23       4027         INX  H              ;POINT NEXT
189B 3600     4028         MVI  M,0            ;SET NEW START
189D 224620   4029         SHLD EXPRS          ;SAVE IT
18A0 3A6821   4030         LDA  DIMSW          ;GET PREV SE
18A3 F5       4031         PUSH PSW            ;SAVE IT
18A4 EB       4032         XCHG                ;RESTORE H,L
18A5 3EFF     4033         MVI  A,255          ;GET ON VALUE
18A7 326821   4034         STA  DIMSW          ;SET SW
18AA CD1C14   4035         CALL LOOKD          ;RECURSIVE CALL
```



```
ADDR OBJECT   LINE TAG     OPCD OPRNDS         COMMENTS

18AD F1       4036         POP  PSW            ;GET DIM SW
18AE 326821   4037         STA  DIMSW          ;REPLACE IT
18B1 221620   4038         SHLD ADDR3          ;SAVE H,L
18B4 E1       4039         POP  H              ;GET EXPRST
18B5 224620   4040         SHLD EXPRS          ;SAVE IT
18B8 E1       4041         POP  H              ;GET STACK COUNTER
18B9 225921   4042         SHLD SPCTR          ;RESTORE IT
18BC F1       4043         POP  PSW            ;GET PAREN COUNT
18BD 325821   4044         STA  PARCT          ;RESTORE IT
18C0 E1       4045         POP  H              ;GET NAM
18C1 D5       4046         PUSH D              ;SAVE ROW
18C2 C5       4047         PUSH B              ;SAVE COL
18C3 EB       4048         XCHG                ;PUT NAME IN D,E
18C4 CD3F1B   4049         CALL SEARC          ;GO FIND ADDRESS (PUT IN D,E)
18C7 D1       4050         POP  D              ;GET ADDR
18C8 C1       4051         POP  B              ;RESTORE COL
18C9 D1       4052         POP  D              ;RESTORE ROW
18CA CDD20A   4053         CALL SUBSC          ;GET SUBSCRIPT (RETURNS ADDR IN
18CD EB       4054         XCHG                ;SAVE IN D,E
18CE 2A1620   4055         LHLD ADDR3          ;GET H,L
18D1 E5       4056         PUSH H              ;SAVE ON STACK
18D2 C36514   4057         JMP  LDV            ;CONTINUE
```

```
ADDR OBJECT   LINE TAG    OPCD OPRNDS          COMMENTS

18D5               TERMI: EQU  $
                   ;
                   ; READ A LINE FROM THE TTY
                   ; FIRST PROMPT WITH THE CHAR IN THE A REG
                   ; TERMINATE THE LINE WITH A X'00'
                   ; IGNORE EMPTY LINES
                   ; CONTROL C WILL CANCEL THE LINE
                   ; CONTROL O WILL TOGGLE THE OUTPUT SWITCH
                   ; RUBOUT WILL DELETE THE LAST CHAR INPUT
                   ;
                   ;
18D5 324420               STA  PROMP           ;SAVE THE PROMPT CHAR
18D8 210420        REIN:  LXI  H,IOBUF         ;POINT TO INPUT BUFFER
18DB 3600                 MVI  M,0             ;MARK BEGIN
18DD 23                   INX  H               ;POINT START
18DE 3A6621               LDA  TAPES           ;GET PT SW
18E1 1F                   RAR                  ;TEST IF LOAD
18E2 DAF518               JC   TREAD           ;BRIF IS
18E5 3A4420               LDA  PROMP           ;GET THE PROMPT AGAIN
18E8 CD2719               CALL TESTO           ;WRITE TO TERMINAL
18EB FE3F                 CPI  '?'             ;TEST IF Q.M.
18ED C2F518               JNZ  TREAD           ;BRIF NOT
18F0 3E20                 MVI  A,' '           ;GET SPACE
18F2 CD2719               CALL TESTO           ;WRITE TO TERMINAL
18F5               TREAD: EQU  $
18F5 DB03                 IN   TTY+1           ;GET TTY STATUS
18F7 E602                 ANI  2               ;TEST IF RXRDY
18F9 CAF518               JZ   TREAD           ;LOOP TILL CHAR
18FC CD131A               CALL GETCH           ;GO READ THE CHAR
18FF 77                   MOV  M,A             ;PUT IN BUFFER
1900 FE0A                 CPI  10              ;TEST IF LINE FEED
1902 CAF518               JZ   TREAD           ;IGNORE IF IT IS
1905 FE0D                 CPI  13              ;TEST IF CR
1907 C24919               JNZ  NOTCR           ;BRIF NOT
190A 3A6621               LDA  TAPES           ;GET PT SW
190D 1F                   RAR                  ;TEST IF LOAD
190E D43219               CNC  CRLF            ;CR/LF IF NOT
1911 3600          CR1:   MVI  M,0             ;MARK END
1913 3A6021               LDA  ILSW            ;GET INPUT LINE SW
1916 B7                   ORA  A               ;TEST IT
1917 C0                   RNZ                  ;RETURN IF ON
1918 2B                   DCX  H               ;POINT PRIOR
1919 7E                   MOV  A,M             ;LOAD IT
191A FE20                 CPI  32              ;TEST IF SPACE
191C CA1119               JZ   CR1             ;BRIF SPACE
191F B7                   ORA  A               ;TEST IF AT BEGINNING
1920 CAD818               JZ   REIN            ;BRIF IS (NULL LINE)
1923 210520               LXI  H,IOBUF         ;POINT BEGIN
1926 C9                   RET                  ;ELSE, RETURN
1927               TESTO: EQU  $
1927 F5                   PUSH PSW             ;SAVE CHAR
```

```
ADDR OBJECT    LINE TAG     OPCD OPRNDS          COMMENTS

1928 DB03      4110 TOUT1:  IN   TTY+1           ;GET STATUS
192A 1F        4111         RAR                  ;TEST IF TXRDY
192B D22819    4112         JNC  TOUT1           ;LOOP TILL READY
192E F1        4113         POP  PSW             ;GET CHAR
192F D302      4114         OUT  TTY             ;WRITE IT
1931 C9        4115         RET                  ;RETURN
1932 3E0D      4116 CRLF:   MVI  A,13            ;GET CR
1934 CD2719    4117         CALL TESTO           ;WRITE IT
1937 3E0A      4118         MVI  A,10            ;LF
1939 CD2719    4119         CALL TESTO           ;WRITE IT
193C 3EFF      4120         MVI  A,255           ;GET RUBOUT CHAR
193E CD2719    4121         CALL TESTO           ;WRITE IT
1941 CD2719    4122         CALL TESTO           ;AGAIN
1944 AF        4123         XRA  A               ;GET A ZERO
1945 326B21    4124         STA  COLUM           ;RESET COLUMN POINTER
1948 C9        4125         RET                  ;RETURN
1949 FE15      4126 NOTCR:  CPI  21              ;TEST IF CONTROL-U
194B C25719    4127         JNZ  NOTCO           ;BRIF NOT
194E CD411A    4128         CALL PRCNT           ;GO PRINT ^U
1951 CD3219    4129         CALL CRLF            ;GET CR/LF
1954 C3D813    4130         JMP  REIN            ;GO RE-ENTER
1957 FE7F      4131 NOTCO:  CPI  127             ;TEST IF RUBOUT
1959 C27A19    4132         JNZ  NOTBS           ;BRIF NOT
195C 3A6621    4133 BSCHA:  LDA  TAPES           ;GET PAPER TAPE SW
195F 1F        4134         RAR                  ;TEST IF LOAD
1960 DAF518    4135         JC   TREAD           ;IGNORE IF LOAD
1963 2B        4136         DCX  H               ;POINT PRIOR
1964 7E        4137         MOV  A,M             ;LOAD PREV CHAR
1965 B7        4138         ORA  A               ;TEST IF BEGIN
1966 CA8519    4139         JZ   ECHO            ;BRIF IS
1969 3E5C      4140         MVI  A,'\'           ;BACKSLASH
196B CD2719    4141         CALL TESTO           ;WRITE IT
196E 7E        4142         MOV  A,M             ;LOAD THE CHAR
196F CD2719    4143         CALL TESTO           ;WRITE IT
1972 3E5C      4144 ATBEG:  MVI  A,'\'           ;BACKSLASH
1974 CD2719    4145         CALL TESTO           ;WRITE IT
1977 C3F518    4146         JMP  TREAD           ;GO READ NEXT
197A 3A6621    4147 NOTBS:  LDA  TAPES           ;GET PAPER TAPE SW
197D 1F        4148         RAR                  ;TEST IF LOAD MODE
197E DA8519    4149         JC   ECHO            ;NOECHO IF IS
1981 7E        4150         MOV  A,M             ;ELSE, LOAD THE CHAR
1982 CD2719    4151 NOTC:   CALL TESTO           ;ECHO THE CHARACTER
1985 23        4152 ECHO:   INX  H               ;POINT NEXT POSIT
1986 C3F518    4153         JMP  TREAD           ;LOOP FOR NEXT
```



```
ADDR OBJECT    LINE TAG     OPCD OPRNDS          COMMENTS

1989           4155 TERMO:  EQU  $
               4156 ;
               4157 ;  TTY PRINT ROUTINE
               4158 ;
               4159 ;  OUTPUT STRING OF CHARS STARTING AT IOBUOO THRU END (FF OR F
               4160 ;  FOLLOWING IMBEDDED CHARACTERS ARE INTERPRETED AS CONTROLS:
               4161 ;  X'00' END OF BUFFER, TYPE CR/LF AND RETURN
               4162 ;  X'FE' END OF BUFFER, RETURN (NO CR/LF)
               4163 ;  X'FD' TYPE CR/LF, CONTINUE
               4164 ;
               4165 ;  RETURN WITHOUT OUTPUT IF OUTPUT SW IS OFF
               4166 ;
               4167 ;
1989 3AF321    4168         LDA  OUTSW           ;GET OUTPUT SW
198C B7        4169         ORA  A               ;TEST IT
198D C0        4170         RNZ                  ;RETURN IF NO PRINT
198E 210520    4171         LXI  H,IOBUF         ;GET ADDR OF BUFFER
1991 7E        4172 OT1:    MOV  A,M             ;LOAD A BYTE
1992 FEFE      4173         CPI  254             ;SEE IF END OF LINE (NO CR/LF)
1994 C8        4174         RZ                   ;RETURN IF EQUAL
1995 FEFD      4175         CPI  253             ;SEE IF IMBEDDED CR/LF
1997 C2A019    4176         JNZ  OT2             ;BRIF NOT
199A CD3219    4177         CALL CRLF            ;LINE FEED
199D C3AF19    4178         JMP  OT4             ;CONTINUE
19A0 B7        4179 OT2:    ORA  A               ;TEST IF END OF OUTPUT
19A1 CA3219    4180         JZ   CRLF            ;BRIF IS
19A4 7E        4181         MOV  A,M             ;LOAD THE BYTE
19A5 CD2719    4182         CALL TESTO           ;TYPE IT
19A8 3A6B21    4183         LDA  COLUM           ;GET COLUMN POINTER
19AB 3C        4184         INR  A               ;ADD ONE
19AC 326B21    4185         STA  COLUM           ;RESTORE IT
19AF 23        4186 OT4:    INX  H               ;POINT NEXT
19B0 C39119    4187         JMP  OT1             ;LOOP
1991           4188 TERMM:  EQU  OT1
```

```
ADDR OBJECT  TAG     OPCD OPRNDS          COMMENTS

19B3         TABST:  EQU  $
             ;
             ;
             ; POSITION TTY AT NEXT TAB STOP
             ;
             ;
19B3 3AF321          LDA  OUTSW           ;GET OUTPUT SWITCH
19B6 B7              ORA  A               ;TEST IT
19B7 C0              RNZ                  ;RETURN IF SUPPRESSED
19B8 3A6B21          LDA  COLUM           ;GET COLUMN POINTER
19BB FE38            CPI  56              ;COMPARE TO 56
19BD D23219          JNC  CRLF            ;BRIF NO ROOM LEFT
19C0 47              MOV  B,A             ;SAVE IT
19C1 AF              XRA  A               ;INIT POSITION
19C2 B8      TBLP:   CMP  B               ;COMPARE
19C3 CAC919          JZ   TBLP2
19C6 D2CE19          JNC  TBON            ;BRIF SHY OF TAB
19C9 C60E    TBLP2:  ADI  14              ;POINT NEXT STOP
19CB C3C219          JMP  TBLP            ;LOOP
19CE 326B21  TBON:   STA  COLUM           ;UPDATE CTR
19D1 90              SUB  B               ;COMPUTE NUMBER OF SPACES
19D2 47              MOV  B,A             ;SAVE IT
19D3 3E20    TBSPA:  MVI  A,' '           ;SPACE TO REG A
19D5 CD2719          CALL TESTO           ;OUTPUT IT
19D8 05              DCR  B               ;SUB 1 FROM CTR
19D9 C8              RZ                   ;RETURN IF ZERO
19DA C3D319          JMP  TBSPA           ;ELSE, LOOP
```



```
ADDR OBJECT  TAG     OPCD OPRNDS          COMMENTS

19DD         LINEO:  EQU  $
             ;
             ; UNPACK LINE NUMBER FROM (H,L) TO (D,E)
             ; ZERO SUPPRESS LEADING ZEROS
             ;
             ;
19DD C5              PUSH B               ;SAVE B,C
19DE 0601            MVI  B,1             ;SET SWITCH
19E0 CDE819          CALL LOUT            ;GO FORMAT 2 BYTES
19E3 CDE819          CALL LOUT            ;THEN THE NEXT 2
19E6 C1              POP  B               ;RESTORE B,C
19E7 C9              RET                  ;RETURN
             ;
19E8         LOUT:   EQU  $
19E8 7E              MOV  A,M             ;GET BYTE
19E9 E6F0            ANI  240             ;ISOLATE LEFT HALF
19EB 1F              RAR                  ;SHIFT RIGHT 1 BIT
19EC 1F              RAR                  ;AGAIN
19ED 1F              RAR                  ;AGAIN
19EE 1F              RAR                  ;LAST TIME
19EF C2F619          JNZ  NOTZ1           ;BRIF NOT ZERO
19F2 B0              ORA  B               ;MERGE IN B
19F3 C2FC19          JNZ  Z1              ;BRIF ZERO
19F6 0600    NOTZ1:  MVI  B,0             ;RESET SWITCH
19F8 F630            ORI  48              ;ZONE
19FA 12              STAX D               ;PUT TO BUFFER
19FB 13              INX  D               ;POINT NEXT
19FC 7E      Z1:     MOV  A,M             ;LOAD BYTE
19FD E60F            ANI  15              ;MASK
19FF C2061A          JNZ  NOTZ2           ;BRIF NOT ZERO
1A02 B0              ORA  B               ;MERGE SWITCH
1A03 C20C1A          JNZ  Z2              ;BRIF ZERO
1A06 0600    NOTZ2:  MVI  B,0             ;SET SWITCH OFF
1A08 F630            ORI  48              ;ZONE
1A0A 12              STAX D               ;PUT TO BUFFER
1A0B 13              INX  D               ;POINT NEXT
1A0C 23      Z2:     INX  H               ;AND NEXT LINE BYTE
1A0D C9              RET                  ;RETURN
```

```
IMSAI 8080 8K BASIC VER=1.3                11:56:42  15 APR 1976  PAGE 129

ADDR OBJECT    LINE TAG    OPCD OPRNDS            COMMENTS

1A0E           4257 TSTCC: EQU  $
               4258 ;
               4259 ;  TEST IF KEY WAS PRESSED DURING EXECUTION
               4260 ;  CANCEL IF CONTROL-C
               4261 ;  TOGGLE OUTPUT SUPPRESS SW IF CONTROL-O
               4262 ;
1A0E DB03      4263         IN   TTY+1              ;GET TTY STATUS
1A10 E602      4264         ANI  2                  ;MASK FOR RXRDY
1A12 C8        4265         RZ                      ;RETURN IF NO CHAR
1A13 DB02      4266 GETCH:  IN   TTY                ;READ THE CHAR
1A15 E67F      4267         ANI  127                ;TURN OFF PARITY
1A17 FE03      4268         CPI  3                  ;TEST IF CONTROL C
1A19 C2321A    4269         JNZ  TSTC1              ;BRIF NOT
1A1C CD411A    4270 CONTC:  CALL PRCNT              ;GO PRINT ^C
1A1F 3A6921    4271         LDA  EDSW               ;GET MODE SW
1A22 B7        4272         ORA  A                  ;TEST IT
1A23 C29101    4273         JNZ  KEY                ;BRIF CMMD MODE
1A26 21D11E    4274 CCRUN:  LXI  H,STOPM            ;POINT MSG
1A29 CD9119    4275         CALL TERMM              ;GO PRINT IT
1A2C CDFC1B    4276         CALL PRLIN              ;GO PRINT LINE
1A2F C39101    4277         JMP  KEY                ;GO TO READY
1A32 FE0F      4278 TSTC1:  CPI  15                 ;TEST IF CONTROL O
1A34 C0        4279         RNZ                     ;RETURN IF NOT
1A35 CD411A    4280 CONTO:  CALL PRCNT              ;GO PRINT ^O
1A38 3AF321    4281         LDA  OUTSW              ;GET OUTPUT SWITCH
1A3B EE01      4282         XRI  1                  ;TOGGLE
1A3D 32F321    4283         STA  OUTSW              ;PUT SW
1A40 C9        4284         RET                     ;RETURN
```

```
IMSAI 8080 8K BASIC VER=1.3                11:58:38  15 APR 1976  PAGE 130

ADDR OBJECT    LINE TAG    OPCD OPRNDS            COMMENTS

1A41           4286 PRCNT:  EQU  $
               4287 ;
               4288 ;
               4289 ;  PRINTS ^ AND CHAR
               4290 ;
1A41 F5        4291         PUSH PSW                ;SAVE CHAR
1A42 3E5E      4292         MVI  A,'^'              ;GET UP ARROW
1A44 CD2719    4293         CALL TESTO              ;WRITE IT
1A47 F1        4294         POP  PSW                ;GET CHAR
1A48 C640      4295         ADI  64                 ;TRNSLATE
1A4A C32719    4296         JMP  TESTO              ;WRITE IT
```

```
ADDR OBJECT   LINE TAG    OPCD OPRNDS            COMMENTS

1A4D               COMP2: EQU  $
                   ;
                   ; CONTINUATION OF COMPARE (RST 2) ROUTINE
                   ;
1A4D B7                   ORA  A                 ;TEST IT
1A4E C2551A               JNZ  COMP5             ;BRIF NOT END
1A51 AF            COMP3: XRA  A                 ;SET EQUAL STATUS
1A52 7E            COMP4: MOV  A,M               ;GET LAST CHAR
1A53 C1                   POP  B                 ;RESTORE B,C
1A54 C9                   RET                    ;RETURN
1A55 BE            COMP5: CMP  M                 ;COMPARE THE TWO CHARS
1A56 CA621A               JZ   COMP6             ;BRIF EQUAL
1A59 78                   MOV  A,B               ;GET COUNT
1A5A FE03                 CPI  3                 ;TEST IF >= 3
1A5C D2511A               JNC  COMP3             ;BRIF NOT LESS THAN 3
1A5F C3521A               JMP  COMP4             ;BRIF LESS THAN 3 AND NOT EQUAL
1A62 04            COMP6: INR  B                 ;COUNT IT
1A63 13                   INX  D                 ;POINT NEXT LIT
1A64 23                   INX  H                 ;POINT NEXT VAR
1A65 C31300               JMP  COMP1             ;CONTINUE
```



```
ADDR OBJECT   LINE TAG    OPCD OPRNDS            COMMENTS

1A68               EOL:   EQU  $
                   ;
                   ; TESTS IF (H,L) IS END OF LINE
                   ; ERROR-DL IF NOT
                   ;
1A68 CF                   RST  1                 ;SKIP TO NON-BLANK
1A69 CD811A               CALL TSTEL             ;TEST IF END LINE
1A6C C2A01A               JNZ  SNERR             ;ERROR IF NOT
1A6F FE3A                 CPI  ':'               ;TEST IF PDP11 MULTI STMT
1A71 CA791A               JZ   EOLO              ;BRIF IS
1A74 FE5C                 CPI  '\'               ;TEST IF MULTIPLE STMT
1A76 C27C1A               JNZ  EOL1              ;BRIF NOT
1A79 326521        EOLO:  STA  MULTI             ;TURN ON SWITCH
1A7C 23            EOL1:  INX  H                 ;POINT NEXT
1A7D 226321               SHLD ENDLI             ;SAVE POINTER
1A80 C9                   RET                    ;RETURN
```

```
ADDR OBJECT    LINE TAG     OPCD OPRNDS             COMMENTS

1A81                 TSTEL:  EQU  $
                     ;
                     ;  TEST (H,L) FOR END OF STMT (00H,'!',':', OR '\')
                     ;  RETURN WITH Z SET IF IS
                     ;
1A81 B7                      ORA  A                 ;TEST FOR ZERO
1A82 C8                      RZ                     ;RETURN IF IS
1A83 FE3A                    CPI  ':'               ;TEST FOR MULTI STMT
1A85 C8                      RZ                     ;RETURN IF IS
1A86 FE5C                    CPI  '\'               ;SAME
1A88 C8                      RZ                     ;RETURN IF IS
1A89 FE21                    CPI  '!'               ;TEST FOR CMMENT
1A8B C9                      RET                    ;RETURN
```



```
ADDR OBJECT    LINE TAG     OPCD OPRNDS             COMMENTS

1A8C                 NOTEO:  EQU  $
                     ;
                     ;
                     ;  TEST IF (H,L) IS END OF LINE
                     ;  RETURN IF NOT, ERROR-DL IF IS
                     ;
1A8C CF                      RST  1                 ;SKIP TO NON-BLANK
1A8D CD811A                  CALL TSTEL             ;TEST IF END OF LINE
1A90 CAA01A                  JZ   SNERR             ;ERROR IF IS
1A93 C9                      RET                    ;ELSE, RETURN
```

```
ADDR OBJECT    LINE TAG     OPCD OPRNDS               COMMENTS

1A94 014C55    4361 ULERR:  LXI  B,'UL'               ;UNDEFINED LINE NUMBER
1A97 DF        4362         RST  3
1A98 01564F    4363 OVERR:  LXI  B,'OV'               ;OVERFLOW/UNDERFLOW/SQR(-X)
1A9B DF        4364         RST  3
1A9C 015453    4365 STERR:  LXI  B,'ST'               ;ERROR IN EXPRESSION STACK
1A9F DF        4366         RST  3
1AA0 014E53    4367 SNERR:  LXI  B,'SN'               ;DELIMITER ERROR
1AA3 DF        4368         RST  3
1AA4 015452    4369 RTERR:  LXI  B,'RT'               ;RETURN & NO GOSUB
1AA7 DF        4370         RST  3
1AA8 014144    4371 DAERR:  LXI  B,'DA'               ;OUT OF DATA
1AAB DF        4372         RST  3
1AAC 01584E    4373 NXERR:  LXI  B,'NX'               ;FOR & NO NEXT / NEXT & NO FOR
1AAF DF        4374         RST  3
1AB0 015643    4375 CVERR:  LXI  B,'CV'               ;CONVERSION ERROR
1AB3 DF        4376         RST  3
```



```
ADDR OBJECT    LINE TAG     OPCD OPRNDS               COMMENTS

1AB4           4378 HDRTL:  EQU  $
               4379 ;
               4380 ;
               4381 ; PUNCH HEADER OR TRAILER ON PAPER TAPE
               4382 ;
               4383 ;
1AB4 0619      4384         MVI  B,25                 ;LOOP COUNT
1AB6 3EFF      4385 HDR1:   MVI  A,255                ;GET RUB OUT CHAR
1AB8 CD2719    4386         CALL TESTO                ;WRITE IT
1ABB 05        4387         DCR  B                    ;DECR LOOP CTR
1ABC C8        4388         RZ                        ;RETURN IF ZERO
1ABD C3B61A    4389         JMP  HDR1                 ;ELSE, LOOP
```

```
ADDR OBJECT   LINE TAG     OPCD OPRNDS          COMMENTS

1AC0          4391 PACK:   EQU  $
              4392 ;
              4393 ; PACK LINE NUMBER FROM (H,L) TO B,C
              4394 ;
              4395 ;
1AC0 010000   4396         LXI  B,0             ;CLEAR B AND C
1AC3 3E04     4397         MVI  A,4             ;INIT DIGIT COUNTER
1AC5 32F421   4398         STA  PRSW            ;SAVE A
1AC8 7E       4399 PK1:    MOV  A,M             ;GET CHAR
1AC9 CD351B   4400         CALL NUMER           ;TEST FOR NUMERIC
1ACC C0       4401         RNZ                  ;RETURN IF NOT NUMERIC
1ACD E60F     4402         ANI  15              ;STRIP OFF ZONE
1ACF 57       4403         MOV  D,A             ;SAVE IT
1AD0 3AF421   4404         LDA  PRSW            ;GET COUNT
1AD3 3D       4405         DCR  A               ;SUBTRACT ONE
1AD4 FAA01A   4406         JM   SNERR           ;BRIF ERROR
1AD7 32F421   4407         STA  PRSW            ;SAVE CTR
1ADA 1E04     4408         MVI  E,4             ;4 BIT SHIFT LOOP
1ADC 79       4409 PK3:    MOV  A,C             ;GET LOW BYTE
1ADD 17       4410         RAL                  ;ROTATE LEFT 1 BIT
1ADE 4F       4411         MOV  C,A             ;REPLACE
1ADF 78       4412         MOV  A,B             ;GET HIGH BYTE
1AE0 17       4413         RAL                  ;ROTATE LEFT 1 BIT
1AE1 47       4414         MOV  B,A             ;REPLACE
1AE2 1D       4415         DCR  E               ;DECR CTR
1AE3 C2DC1A   4416         JNZ  PK3             ;LOOP
1AE6 79       4417         MOV  A,C             ;GET LOW
1AE7 B2       4418         ORA  D               ;PUT DIGIT IN RIGHT HALF OF BYTE
1AE8 4F       4419         MOV  C,A             ;REPLACE
1AE9 23       4420         INX  H               ;POINT NEXT BYTE
1AEA C3C31A   4421         JMP  PK1             ;LOOP
```



```
ADDR OBJECT   LINE TAG     OPCD OPRNDS          COMMENTS

1AED          4423 SQUIS:  EQU  $
              4424 ;
              4425 ; COMPRESSES THE EXPR STACK
              4426 ; REG A CONTAINS NUMBER OF BYTES TO REMOVE STARTING AT (H,L+1
              4427 ; CONTAINS TOTAL NUBER OF CHARACTERS IN STACK THUS FAR
              4428 ;
1AED E5       4429         PUSH H               ;SAVE H,L
1AEE 5F       4430         MOV  E,A             ;COUNT TO E
1AEF 1600     4431         MVI  D,0             ;ZERO HI BYTE
1AF1 19       4432         DAD  D               ;COMPUTE START
1AF2 EB       4433         XCHG                 ;PUT TO D,E
1AF3 E1       4434         POP  H               ;GET H,L
1AF4 2F       4435         CMA                  ;COMPLEMENT COUNT
1AF5 3C       4436         INR  A               ;THEN 2'S COMPLEMENT
1AF6 80       4437         ADD  B               ;COMPUTE B-A
1AF7 47       4438         MOV  B,A             ;PUT TO B
1AF8 13       4439 SQUI2:  INX  D               ;POINT NEXT SEND
1AF9 23       4440         INX  H               ;POINT NEXT RECEIVE
1AFA 1A       4441         LDAX D               ;GET A CHAR
1AFB 77       4442         MOV  M,A             ;PUT IT DOWN
1AFC 05       4443         DCR  B               ;DECR CTR
1AFD C2F81A   4444         JNZ  SQUI2           ;LOOP
1B00 224620   4445         SHLD EXPRS           ;UPDATE NEW START OF EXPR
1B03 C9       4446         RET                  ;RETURN
```

```
ADDR OBJECT   LINE TAG     OPCD OPRNDS            COMMENTS

1B04          4448 SKP2Z:  EQU  $
              4449 ;
              4450 ; FIND END OF LITERAL IN (D,E)
              4451 ;
1B04 1A       4452         LDAX D                 ;GET BYTE OF LIT
1B05 B7       4453         ORA  A                 ;TEST IT
1B06 C8       4454         RZ                     ;RETURN IF ZERO (END).
1B07 13       4455         INX  D                 ;ELSE, POINT NEXT
1B08 C3041B   4456         JMP  SKP2Z             ;LOOP
```



```
ADDR OBJECT   LINE TAG     OPCD OPRNDS            COMMENTS

1B0B          4458 GTEMP:  EQU  $
              4459 ;
              4460 ; GETS FOUR BYTE TEMPORARY STORAGE AREA,
              4461 ; STORES THE FACC THERE,
              4462 ; PUTS ADDR OF AREA IN EXPR STACK (H,L)
              4463 ;
1B0B EB       4464         XCHG                   ;SAVE H,L IN D,E
1B0C E3       4465         XTHL                   ;EXCHANGE 0 AND RET ADDR
1B0D E5       4466         PUSH H                 ;PUT NEW RET ADDR
1B0E E5       4467         PUSH H                 ;DOIT IT AGAIN
1B0F 210000   4468         LXI  H,0               ;ZERO H,L
1B12 39       4469         DAD  SP                ;GET SP ADDR IN H,L
1B13 23       4470         INX  H                 ;PLUS ONE
1B14 23       4471         INX  H                 ;PLUS ONE MORE (POINT TO NEW ARE
1B15 C5       4472         PUSH B                 ;SAVE CTRS
1B16 D5       4473         PUSH D                 ;SAVE EXPR ADDR
1B17 E5       4474         PUSH H                 ;SAVE TEMP ADDR
1B18 F7       4475         RST  6                 ;GO STORE FACC
1B19 D1       4476         POP  D                 ;RESTORE TEMP ADDR
1B1A 2A5921   4477         LHLD SPCTR             ;GET COUNT
1B1D 23       4478         INX  H                 ;PLUS ONE
1B1E 23       4479         INX  H                 ;ONE MORE
1B1F 225921   4480         SHLD SPCTR             ;PUT BACK
1B22 E1       4481         POP  H                 ;RESTORE EXPR ADDR
1B23 C1       4482         POP  B                 ;RESTORE CTRS
1B24 23       4483 SADR:   INX  H                 ;POINT NEXT BYTE
1B25 72       4484         MOV  M,D               ;HIGH BYTE TO EXPRSTK
1B26 23       4485         INX  H                 ;POINT NEXT
1B27 73       4486         MOV  M,E               ;LOW BYTE TO EXPR STK
1B28 23       4487         INX  H                 ;POINT NEXT
1B29 36E3     4488         MVI  M,227             ;CODE = NUMERIC DATA
1B2B C9       4489         RET                    ;RETURN
```

```
ADDR OBJECT    LINE TAG     OPCD OPRNDS            COMMENTS

1B2C           4491 ALPHA:  EQU  $
               4492 ;
               4493 ;  TESTS THE CHAR AT (H,L)
               4494 ;  RETURNS WITH Z SET IF CHAR IS ALPHA (A-Z)
               4495 ;  RETURNS WITH Z OFF IF NOT ALPHA
               4496 ;  CHAR IS LEFT IN REG A
               4497 ;
1B2C 7E        4498         MOV  A,M               ;PUT CHAR TO REG A
1B2D FE41      4499         CPI  'A'               ;TEST IF A OR HIGHER
1B2F D8        4500         RC                     ;RETURN IF NOT ALPHA (Z IS OFF)
1B30 FE5A      4501         CPI  'Z'               ;TEST IF Z OR LESS
1B32 C33B1B    4502         JMP  NUMEN             ;GO WRAPUP
```



```
ADDR OBJECT    LINE TAG     OPCD OPRNDS            COMMENTS

1B35           4504 NUMER:  EQU  $
               4505 ;
               4506 ;  TESTS THE CHAR AT (H,L)
               4507 ;  RETURNS WITH Z SET IF NUMERIC (0-9)
               4508 ;  ELSE Z IS OFF
               4509 ;  CHAR IS LEFT IN THE A REG
               4510 ;
1B35 7E        4511         MOV  A,M               ;GET CHAR TO REG A
1B36 FE30      4512         CPI  '0'               ;TEST IF ZERO OR GREATER
1B38 D8        4513         RC                     ;RETURN IF LESS THAN ZERO
1B39 FE39      4514         CPI  '9'               ;TEST IF 9 OR LESS
1B3B C8        4515 NUMEN:  RZ                     ;RETURN IF 9
1B3C D0        4516         RNC                    ;RETURN IF NOT NUMERIC
1B3D BF        4517         CMP  A                 ;SET Z
1B3E C9        4518         RET                    ;RETURN
```

```
ADDR OBJECT   LINE TAG     OPCD OPRNDS          COMMENTS

1B3F          4520 SEARC:  EQU  $
              4521 ;
              4522 ; SEARCHES FOR THE VARIABLE IN D,E
              4523 ; RETURNS WITH ADDR OF DATA AREA FOR VARIABLE
              4524 ;
1B3F E5       4525         PUSH H               ;SAVE H,L
1B40 3A7D21   4526         LDA  FNMOD           ;GET FUNCTION MODE
1B43 B7       4527         ORA  A               ;TEST IT
1B44 C29A1B   4528         JNZ  SCH6            ;BRIF IN A FUNCTION
1B47 2AF321   4529 SCH0:   LHLD DATAB           ;GET ADDR OF DATA POOL
1B4A 7E       4530 SCH1:   MOV  A,M             ;GET THE BYTE
1B4B B7       4531         ORA  A               ;TEST IF END
1B4C CA701B   4532         JZ   SCH3            ;BRIF END
1B4F 2B       4533         DCX  H               ;POINT NEXT
1B50 2B       4534         DCX  H               ;DITTO
1B51 46       4535         MOV  B,M             ;GET HI LEN
1B52 2B       4536         DCX  H               ;POINT NEXT
1B53 4E       4537         MOV  C,M             ;GET LO LEN
1B54 FF       4538         RST  7               ;ADJUST H,L
1B55 03       4539         DB   3
1B56 7E       4540         MOV  A,M             ;LOAD 1ST CHAR
1B57 BA       4541         CMP  D               ;COMPARE 1ST CHAR
1B58 C26C1B   4542         JNZ  SCH2            ;BRIF NOT EQUAL
1B5B 2B       4543         DCX  H               ;POINT NEXT
1B5C 7E       4544         MOV  A,M             ;LOAD 2ND DIGIT
1B5D 23       4545         INX  H               ;POINT BACK
1B5E BB       4546         CMP  E               ;COMPARE 2ND CHAR
1B5F C26C1B   4547         JNZ  SCH2            ;BRIF NOT EQUAL
1B62 7A       4548         MOV  A,D             ;GET HI NAME
1B63 B7       4549         ORA  A               ;TEST IT
1B64 FACF1B   4550         JM   SCH9            ;RETURN IF MATRIX
1B67 09       4551         DAD  B               ;POINT NEXT ENTRY
1B68 23       4552         INX  H               ;PLUS ONE
1B69 EB       4553         XCHG                 ;FLIP/FLOP
1B6A E1       4554         POP  H               ;RESTORE H
1B6B C9       4555         RET                  ;RETURN
1B6C 09       4556 SCH2:   DAD  B               ;MINUS LEN
1B6D C34A1B   4557         JMP  SCH1            ;LOOP
1B70 72       4558 SCH3:   MOV  M,D             ;PUT 1ST CHAR
1B71 2B       4559         DCX  H               ;POINT NEXT
1B72 73       4560         MOV  M,E             ;PUT 2ND CHAR
1B73 2B       4561         DCX  H               ;POINT NEXT
1B74 7A       4562         MOV  A,D             ;GET HI NAME
1B75 B7       4563         ORA  A               ;TEST IT
1B76 FAAE1B   4564         JM   SCH7            ;BRIF ARRAY
1B79 36FF     4565         MVI  M,255           ;HI LEN
1B7B 2B       4566         DCX  H               ;POINT NEXT
1B7C 7B       4567         MOV  A,E             ;GET LO NAME
1B7D B7       4568         ORA  A               ;TEST TYPE
1B7E FA881B   4569         JM   SCH4            ;BRIF CHAR
1B81 36F8     4570         MVI  M,248           ;LO LEN
```

```
ADDR OBJECT   LINE TAG    OPCD OPRNDS         COMMENTS

1B83 0604     4571        MVI  B,4            ;LOOP CTR
1B85 C38C1B   4572        JMP  SCH5           ;BRARND
1B88 36FB     4573 SCH4:  MVI  M,251          ;LO LEN
1B8A 0601     4574        MVI  B,1            ;LOOP CTR
1B8C 2B       4575 SCH5:  DCX  H              ;POINT NEXT
1B8D 3600     4576        MVI  M,0            ;ZERO THE VALUE
1B8F 05       4577        DCR  B              ;DECR CTR
1B90 C28C1B   4578        JNZ  SCH5           ;LOOP
1B93 2B       4579        DCX  H              ;POINT NEXT
1B94 3600     4580        MVI  M,0            ;MARK NEW END
1B96 23       4581        INX  H              ;POINT ADDR OF VARIABLE
1B97 EB       4582        XCHG                ;PUT LOCATION TO D,E
1B98 E1       4583        POP  H              ;RESTORE H,L
1B99 C9       4584        RET                 ;RETURN
1B9A 215C21   4585 SCH6:  LXI  H,FNARG        ;POINT DUMMY ARG
1B9D 7E       4586        MOV  A,M            ;LOAD 1ST CHAR
1B9E BA       4587        CMP  D              ;COMPARE
1B9F C2471B   4588        JNZ  SCH0           ;BRIF NOT EQUAL
1BA2 23       4589        INX  H              ;POINT NEXT
1BA3 7E       4590        MOV  A,M            ;LOAD 2ND CHAR
1BA4 BB       4591        CMP  E              ;COMPARE
1BA5 C2471B   4592        JNZ  SCH0           ;BRIF NOT EQUAL
1BA8 23       4593        INX  H              ;POINT NEXT
1BA9 56       4594        MOV  D,M            ;GET HI ADDR
1BAA 23       4595        INX  H              ;POINT NEXT
1BAB 5E       4596        MOV  E,M            ;GET LO ADDR
1BAC E1       4597        POP  H              ;RESTORE H,L
1BAD C9       4598        RET                 ;RETURN
1BAE E5       4599 SCH7:  PUSH H              ;SAVE ADDRESS
1BAF 36FE     4600        MVI  M,254          ;MOVE HI DISP
1BB1 2B       4601        DCX  H              ;POINT NEXT
1BB2 3614     4602        MVI  M,20           ;MOVE LO DISP
1BB4 2B       4603        DCX  H
1BB5 3600     4604        MVI  M,0            ;MOVE A ZERO
1BB7 2B       4605        DCX  H              ;POINT NEXT
1BB8 360A     4606        MVI  M,10           ;MOVE 10
1BBA 2B       4607        DCX  H              ;POINT NEXT
1BBB 3600     4608        MVI  M,0            ;MOVE A ZERO
1BBD 2B       4609        DCX  H              ;POINT NEXT
1BBE 360A     4610        MVI  M,10           ;MOVE A 10 (DEFAULT IS 10 X 10)
1BC0 01E501   4611        LXI  B,485          ;TOTAL NUMBER OF BYTES TAKEN BY
1BC3 2B       4612 SCH8:  DCX  H              ;POINT NEXT
1BC4 3600     4613        MVI  M,0            ;CLEAR ONE BYTE
1BC6 0B       4614        DCX  B              ;DCR CTR
1BC7 78       4615        MOV  A,B            ;GET HI
1BC8 B1       4616        ORA  C              ;PLUS LO
1BC9 C2C31B   4617        JNZ  SCH8           ;LOOP
1BCC E1       4618        POP  H              ;RESTORE PTR TO START
1BCD 23       4619        INX  H              ;POINT LO NAME
1BCE 23       4620        INX  H              ;POINT HI NAME
1BCF C1       4621 SCH9:  POP  B              ;NEED TO EXCHANGE LAST TWO ENTRI.
```



```
ADDR OBJECT   LINE TAG    OPCD OPRNDS         COMMENTS

1BD0 D1       4622        POP  D              ;SO DOIT
1BD1 C5       4623        PUSH B
1BD2 D5       4624        PUSH D
1BD3 C9       4625        RET                 ;RETURN
```

```
ADDR OBJECT    LINE TAG     OPCD OPRNDS            COMMENTS

1BD4                VAR:    EQU  $
                    ;
                    ;
                    ; TEST (H,L) FOR A VARIABLE NAME
                    ; PUTS THE NAME IN D,E IF FOUND
                    ; ERROR SN IF NONE FOUND
                    ;
1BD4 CF                     RST  1                 ;SKIP TO NON-BLANK
1BD5 CD2C1B                 CALL ALPHA             ;TEST IF ALPHA
1BD8 C2A01A                 JNZ  SNERR             ;BRIF NOT ALPHA
1BDB 57                     MOV  D,A               ;FIRST CHAR
1BDC 1E20                   MVI  E,' '             ;DEFAULT
1BDE 23                     INX  H                 ;POINT NEXT
1BDF CF                     RST  1                 ;GET 2ND CHAR
1BE0 CD351B                 CALL NUMER             ;TEST IF NUMERIC
1BE3 C2E91B                 JNZ  VAR2              ;BRIF NOT NUMERIC
1BE6 5F                     MOV  E,A               ;SAVE 2ND CHAR
1BE7 23                     INX  H                 ;POINT NEXT
1BE8 CF                     RST  1                 ;GET NON-BLANK FOLLOWING
1BE9 FE24           VAR2:   CPI  '$'               ;TEST IF STRING
1BEB C2F41B                 JNZ  VAR3              ;BRIF NOT
1BEE 7B                     MOV  A,E               ;GET 2ND CHAR
1BEF F680                   ORI  128               ;SET TYPE
1BF1 5F                     MOV  E,A               ;SAVE IT
1BF2 23                     INX  H                 ;SKIP $
1BF3 C9                     RET                    ;THEN RETURN
1BF4 FE28           VAR3:   CPI  '('               ;TEST IF ARRAY
1BF6 C0                     RNZ                    ;RETURN IF NOT
1BF7 7A             VAR4:   MOV  A,D               ;GET HI NAME
1BF8 F680                   ORI  128               ;TURN ON D7
1BFA 57                     MOV  D,A               ;RESTORE
1BFB C9                     RET                    ;RETURN
```

IMSAI 8080 8K BASIC VER=1.3 12:24:04 15 APR 1976 PAGE 147

```
ADDR OBJECT    LINE TAG     OPCD OPRNDS            COMMENTS

1BFC                PRLIN:  EQU  $
                    ;
                    ; PRINTS LINE NUMBER FOLLOWED BY CR/LF
                    ;
1BFC 110020                 LXI  D,LINEN           ;POINT AREA
1BFF 2A7E21                 LHLD LINE              ;GET ADDR OF LINE NUMBER
1C02 CDDD19                 CALL LINEO             ;GO UNPACK
1C05 EB                     XCHG                   ;PUT TO H,L
1C06 3600                   MVI  M,0               ;END OF MSG
1C08 210020                 LXI  H,LINEN           ;POINT AREA
1C0B C39119                 JMP  TERMM             ;GO PRINT IT
```

```
ADDR OBJECT    LINE TAG     OPCD OPRNDS            COMMENTS

1C0E           4672 ERROR:  EQU  $
               4673 ;
               4674 ;  CONTINUATION OF ERROR MSG FORMATER (RST 3)
               4675 ;
               4676 ;
1C0E 71        4677         MOV  M,C               ;PUT 2ND CHAR
1C0F 23        4678         INX  H                 ;POINT NEXT
1C10 36FE      4679         MVI  M,254             ;MARK END
1C12 210520    4680         LXI  H,IOBUF           ;POINT
1C15 CD9119    4681         CALL TERMM             ;PRINT XX
1C18 21E01E    4682         LXI  H,ERRMS           ;POINT MSG
1C1B CD9119    4683         CALL TERMM             ;PRINT "ERROR IN LINE "
1C1E CDFC1B    4684         CALL PRLIN             ;GO PRINT LINE
1C21 C39800    4685         JMP  RDY               ;GOTO READY
```



```
ADDR OBJECT    LINE TAG     OPCD OPRNDS            COMMENTS

1C24           4687 COPYD:  EQU  $
               4688 ;
               4689 ;  MOVE THE STRING FROM (D,E) TO (H,L) COUNT IN B
               4690 ;
1C24 1A        4691         LDAX D                 ;GET A BYTE
1C25 77        4692         MOV  M,A               ;MOVE IT
1C26 23        4693         INX  H                 ;POINT NEXT
1C27 13        4694         INX  D                 ;DITTO
1C28 05        4695         DCR  B                 ;DECR CTR
1C29 C2241C    4696         JNZ  COPYD             ;LOOP
1C2C C9        4697         RET                    ;THEN RETURN
```

```
ADDR OBJECT   LINE TAG     OPCD OPRNDS            COMMENTS

1C2D          4699 COPYH:  EQU  $
              4700 ;
              4701 ; MOVE THE STRING FROM (H,L) TO (D,E) COUNT IN B
              4702 ;
1C2D EB       4703         XCHG                   ;FLIP/FLOP
1C2E CD241C   4704         CALL COPYD             ;GO COPY
1C31 EB       4705         XCHG                   ;FLIP/FLOP BACK
1C32 C9       4706         RET                    ;RETURN
```



```
ADDR OBJECT   LINE TAG     OPCD OPRNDS            COMMENTS

1C33          4708 ZEROM:  EQU  $
              4709 ;
              4710 ; MOVES A STRING OF BINARY ZEROS, COUNT IN B
              4711 ;
1C33 3600     4712         MVI  M,0               ;MOVE A ZERO
1C35 23       4713         INX  H                 ;POINT NEXT
1C36 05       4714         DCR  B                 ;DECR CTR
1C37 C2331C   4715         JNZ  ZEROM             ;LOOP
1C3A C9       4716         RET                    ;RETURN
```

```
ADDR OBJECT    LINE TAG    OPCD OPRNDS            COMMENTS

1C3B                FBIN:  EQU  $
                    ;
                    ;
                    ; CONVERT FLOAT ACC TO UNSIGNED BINARY NUMBER IN A REG
                    ; RETURNS 0 IN A REG IF FACC<0 OR FACC>255
                    ;
                    ;
1C3B E5                    PUSH H                 ;SAVE H,L
1C3C CD4511                CALL INT               ;GO CONVERT TO INTEGER
1C3F FA601C                JM   FBIN2             ;BRIF NEGATIVE
1C42 3A4820                LDA  FACC              ;GET EXPONENT
1C45 CD830F                CALL FEXP              ;EXPAND TO 8 BITS
1C48 FE09                  CPI  9                 ;TEST FOR MAX
1C4A D2601C                JNC  FBIN2             ;BRIF TOO BIG
1C4D 47                    MOV  B,A               ;SAVE EXPONENT
1C4E 3E08                  MVI  A,8               ;GET MAX
1C50 90                    SUB  B                 ;MINUS COUNT
1C51 47                    MOV  B,A               ;SAVE SHIFT COUNT
1C52 3A4920                LDA  FACC+1            ;GET MSB
1C55 CA5E1C                JZ   FBIN1             ;BRIF 8 BITS
1C58 B7             FBIN0: ORA  A                 ;TURN OFF CY
1C59 1F                    RAR                    ;SHIFT
1C5A 05                    DCR  B                 ;DECR CTR
1C5B C2581C                JNZ  FBIN0             ;LOOP
1C5E E1             FBIN1: POP  H                 ;RESTORE H,L
1C5F C9                    RET                    ;RETURN
1C60 AF             FBIN2: XRA  A                 ;GET A ZERO
1C61 E1                    POP  H                 ;RESTORE H,L
1C62 C9                    RET                    ;RETURN
```



```
ADDR OBJECT    LINE TAG    OPCD OPRNDS            COMMENTS

1C63                ARG:   EQU  $
                    ;
                    ; GET NEXT ARGUMENT FROM POLISH STACK
                    ;
1C63 2A1220                LHLD ADDR1             ;GET ADDRESS
1C66 23                    INX  H                 ;POINT NEXT
1C67 56                    MOV  D,M               ;GET HI ADDRESS
1C68 23                    INX  H                 ;POINT NEXT
1C69 5E                    MOV  E,M               ;GET LO ADDRESS
1C6A 23                    INX  H                 ;POINT TYPE
1C6B 221220                SHLD ADDR1             ;GET ADDRESS
1C6E 2B                    DCX  H                 ;POINT BACK
1C6F C32613                JMP  EVLD              ;CALL EVLOAD AND RETURN
                    ;
                    ;
1C72                ARGNU: EQU  $
                    ;
1C72 CD631C                CALL ARG               ;GET ARGUMENT
1C75 C33B1C                JMP  FBIN              ;THEN CONVERT FACC TO BIN
```

```
ADDR OBJECT   LINE TAG    OPCD OPRNDS          COMMENTS

1C78               FN:    EQU  $
                   ;
                   ; STMT: DEF FNX(A)=EXPR
                   ;
                   ; NOTE: ENTRY FROM EXPR ANALYZER (RECURSIVE)
                   ;
1C78 C5                   PUSH B               ;SAVE B,C
1C79 D5                   PUSH D               ;SAVE D,E
1C7A E5                   PUSH H               ;SAVE H,L
1C7B EB                   XCHG                 ;PUT H,L TO D,E
1C7C 2A1620               LHLD ADDR3           ;GET ADDR
1C7F E5                   PUSH H               ;SAVE IT
1C80 EB                   XCHG                 ;PUT D,E BACK TO H,L
1C81 221620               SHLD ADDR3           ;UPDATE PTR
1C84 2A5921               LHLD SPCTR           ;GET SP COUNT
1C87 E5                   PUSH H               ;SAVE IT
1C88 3A5821               LDA  PARCT           ;GET PAREN COUNT
1C8B 47                   MOV  B,A             ;PUT TO B
1C8C 3A7D21               LDA  FNMOD           ;GET FN MODE
1C8F 4F                   MOV  C,A             ;PUT TO C
1C90 C5                   PUSH B               ;SAVE B,C
1C91 3A6821               LDA  DIMSW           ;GET DIM SW
1C94 F5                   PUSH PSW             ;SAVE IT
1C95 AF                   XRA  A               ;CLEAR A
1C96 326821               STA  DIMSW           ;RESET DIM SW
1C99 2A5C21               LHLD FNARG           ;GET OLD ARG NAME
1C9C E5                   PUSH H               ;SAVE
1C9D 2A5E21               LHLD FNARG+2         ;GET OLD ARG ADDRESS
1CA0 E5                   PUSH H               ;SAVE
1CA1 2AFA21               LHLD PROGE           ;GET END OF PROGRAM
1CA4 E5                   PUSH H               ;SAVE IT
1CA5 2A4620               LHLD EXPRS           ;GET END OF EXPR
1CA8 E5                   PUSH H               ;SAVE IT
1CA9 22FA21               SHLD PROGE           ;SAVE NEW 'END' OF PROGRAM
1CAC 3E01                 MVI  A,1             ;GET ON SETTING
1CAE 327D21               STA  FNMOD           ;SET IN FUNCTION
1CB1 2A1620               LHLD ADDR3           ;POINT TO EXPR
1CB4 4E                   MOV  C,M             ;GET FN CHAR
1CB5 2B                   DCX  H               ;POINT BACK
1CB6 46                   MOV  B,M             ;GET HI NAME
1CB7 214522               LXI  H,BEGPR         ;POINT START OF PROGRAM
1CBA 7E            FN2:   MOV  A,M             ;LOAD LEN TO NEXT STMT
1CBB B7                   ORA  A               ;TEST IF AT END
1CBC CAA01A               JZ   SNERR           ;BRIF FN NOT FOUND
1CBF E5                   PUSH H               ;SAVE PTR
1CC0 FF                   RST  7               ;ADJUST H,L
1CC1 03                   DB   3
1CC2 11BB1F               LXI  D,DEFLI         ;LITERAL
1CC5 D7                   RST  2               ;GO COMPARE
1CC6 C2D81C               JNZ  FN3             ;BRIF NOT EQUAL
1CC9 C5                   PUSH B               ;SAVE TEST NAME
```

```
ADDR OBJECT   LINE TAG   OPCD OPRNDS      COMMENTS

1CCA CDD41B   4819       CALL VAR         ;GO GET NAME
1CCD C1       4820       POP  B           ;RESTORE NAME
1CCE 7A       4821       MOV  A,D         ;GET HI NAME
1CCF B8       4822       CMP  B           ;COMPARE
1CD0 C2D81C   4823       JNZ  FN3         ;BRIF NOT EQUAL
1CD3 7B       4824       MOV  A,E         ;GET LO
1CD4 B9       4825       CMP  C           ;COMPARE
1CD5 CAE01C   4826       JZ   FN4         ;BRIF EQUAL
1CD8 E1       4827 FN3:  POP  H           ;GET OLD PTR
1CD9 5E       4828       MOV  E,M         ;GET LO LEN
1CDA 1600     4829       MVI  D,0         ;ZERO HI LEN
1CDC 19       4830       DAD  D           ;POINT NEXT STMT
1CDD C3BA1C   4831       JMP  FN2         ;LOOP
1CE0 D1       4832 FN4:  POP  D           ;ADJUST STACK
1CE1 CF       4833       RST  1           ;SKIP BLANKS
1CE2 FE28     4834       CPI  '('         ;TEST IF OPEN PAREN
1CE4 C2A01A   4835       JNZ  SNERR       ;BRIF NOT
1CE7 23       4836       INX  H           ;SKIP IT
1CE8 CDD41B   4837       CALL VAR         ;GO GET VAR NAME
1CEB E5       4838       PUSH H           ;SAVE H ADDR
1CEC 215C21   4839       LXI  H,FNARG     ;POINT DUMMY ARG TBL
1CEF 72       4840       MOV  M,D         ;STORE LETTER
1CF0 23       4841       INX  H           ;POINT NEXT
1CF1 73       4842       MOV  M,E         ;STORE DIGIT
1CF2 23       4843       INX  H           ;POINT NEXT
1CF3 EB       4844       XCHG             ;PUT H,L TO D,E
1CF4 2A1620   4845       LHLD ADDR3       ;POINT TO EXPR STACK
1CF7 23       4846       INX  H           ;POINT CODE
1CF8 23       4847       INX  H           ;POINT HI ADR
1CF9 7E       4848       MOV  A,M         ;GET HI
1CFA 12       4849       STAX D           ;PUT TO TABLE
1CFB 13       4850       INX  D           ;POINT NEXT
1CFC 23       4851       INX  H           ;DITTO
1CFD 7E       4852       MOV  A,M         ;GET LO ADDR
1CFE 12       4853       STAX D           ;PUT TO TABLE
1CFF E1       4854       POP  H           ;RESTORE PTR TO STMT
1D00 CF       4855       RST  1           ;SKIP BLANKS
1D01 FE29     4856       CPI  ')'         ;TEST IF CLOSE PAREN
1D03 C2A01A   4857       JNZ  SNERR       ;BRIF NOT
1D06 23       4858       INX  H           ;SKIP IT
1D07 CF       4859       RST  1           ;SKIP BLANKS
1D08 FE3D     4860       CPI  '='         ;TEST IF EQUAL SIGN
1D0A C2A01A   4861       JNZ  SNERR       ;BRIF NOT
1D0D 23       4862       INX  H           ;SKIP IT
1D0E CD0714   4863       CALL EXPR        ;GO EVAL FUNCTION
1D11 CD681A   4864       CALL EOL         ;MUST BE END OF LINE
1D14 E1       4865       POP  H           ;GET H,L
1D15 224620   4866       SHLD EXPRS       ;RESTORE START OF EXPR
1D18 E1       4867       POP  H           ;GET H,L
1D19 22FA21   4868       SHLD PROGE       ;RESTORE 'END' OF PROGRAM
1D1C E1       4869       POP  H           ;GET H,L
```



```
ADDR OBJECT   LINE TAG   OPCD OPRNDS      COMMENTS

1D1D 225E21   4870       SHLD FNARG+2     ;STORE ADDR
1D20 E1       4871       POP  H           ;GET H,L
1D21 225C21   4872       SHLD FNARG       ;STORE DUMMY ARG
1D24 F1       4873       POP  PSW         ;GET A,STATUS
1D25 326821   4874       STA  DIMSW       ;RESTORE DIM SW
1D28 C1       4875       POP  B           ;GET B,C
1D29 79       4876       MOV  A,C         ;LOAD C
1D2A 327D21   4877       STA  FNMOD       ;RESTORE MOE
1D2D 78       4878       MOV  A,B         ;LOAD B
1D2E 325821   4879       STA  PARCT       ;RESTORE PAREN COUNT
1D31 E1       4880       POP  H           ;GET H,L
1D32 225921   4881       SHLD SPCTR       ;RESTORE SP COUNTER
1D35 E1       4882       POP  H           ;GET H,L
1D36 221620   4883       SHLD ADDR3       ;RESTORE ADDR OF EVAL
1D39 E1       4884       POP  H           ;GET H,L
1D3A D1       4885       POP  D           ;GET D,E
1D3B 2B       4886       DCX  H           ;POINT 2ND BYTE FOLLOWING OP
1D3C 221420   4887       SHLD ADDR2       ;SAVE IT
1D3F FF       4888       RST  7           ;POINT TO ARG TYPE
1D40 05       4889       DB   5
1D41 221220   4890       SHLD ADDR1       ;SAVE ADDR
1D44 C3A816   4891       JMP  EV3         ;GO WRAPUP
```

```
IMSAI 8080 8K BASIC VER=1.3                          12:45:02  15 APR 1976  PAGE 157

ADDR OBJECT    LINE TAG     OPCD OPRNDS              COMMENTS

1D47           4893 BINFL:  EQU  $
               4894 ;
               4895 ; CONVERT D,E TO FLOATING POINT NUMBER IN FAC
               4896 ;
               4897 ;
1D47 214820    4898         LXI  H,FACC             ;POINT ACC
1D4A 3618      4899         MVI  M,24               ;MAX BITS
1D4C 23        4900         INX  H                  ;POINT NEXT
1D4D 3600      4901         MVI  M,0                ;CLEAR MSB
1D4F 23        4902         INX  H                  ;POINT NEXT
1D50 72        4903         MOV  M,D                ;MOVE MID
1D51 23        4904         INX  H                  ;POINT NEXT
1D52 73        4905         MOV  M,E                ;MOVE LSB
1D53 C31D0E    4906         JMP  FNORM              ;GO NORMALIZE & RETURN


IMSAI 8080 8K BASIC VER=1.3                          12:46:24  15 APR 1976  PAGE 158

ADDR OBJECT    LINE TAG     OPCD OPRNDS              COMMENTS

1D56 3FFD      4908 RNDP:   DB   16381              ;16381
1D58 3FEB      4909 RNDQ:   DB   16363              ;16363
1D5A 3FDD      4910 RNDR:   DB   16349              ;16349
1D5C 1BEC      4911 NRNDX:  DB   7148
1D5E 33D3      4912 NRNDY:  DB   13267
1D60 1A85      4913 NRNDZ:  DB   6789
1D62 2B1E      4914 NRNDS:  DB   11038
1D64 57484154  4915 WHATL:  DB   'WHAT',0
1D68 00
1D69 4C494E45  4916 LLINE:  DB   'LINE',0
1D6D 00
1D6E 544142    4917 TABLI:  DB   'TAB',0
1D71 00
1D72           4918 FUNCT:  EQU  $
1D72 414253    4919 ABSLI:  DB   'ABS',0
1D75 00
1D76 2A11      4920         DW   ABS
1D78 AB        4921         DB   171
1D79 535152    4922 SQRLI:  DB   'SQR',0
1D7C 00
1D7D 8A11      4923         DW   SQR
1D7F AB        4924         DB   171
1D80 494E54    4925 INTLI:  DB   'INT',0
1D83 00
1D84 4511      4926         DW   INT
1D86 AB        4927         DB   171
1D87 53474E    4928 SGNLI:  DB   'SGN',0
1D8A 00
1D8B 3311      4929         DW   SGN
1D8D AB        4930         DB   171
1D8E 524E44    4931 RNDLI:  DB   'RND',0
1D91 00
1D92 E411      4932         DW   RND
1D94 AB        4933         DB   171
1D95 53494E    4934 SINLI:  DB   'SIN',0
1D98 00
1D99 A90F      4935         DW   SIN
1D9B AB        4936         DB   171
1D9C 434F53    4937 COSLI:  DB   'COS',0
1D9F 00
1DA0 3110      4938         DW   COS
1DA2 AB        4939         DB   171
1DA3 54414E    4940 TANLI:  DB   'TAN',0
1DA6 00
1DA7 3A10      4941         DW   TAN
1DA9 AB        4942         DB   171
1DAA 41544E    4943 ATNLI:  DB   'ATN',0
1DAD 00
1DAE 5E10      4944         DW   ATN
1DB0 AB        4945         DB   171
1DB1 494E50    4946 INPLI:  DB   'INP',0
```

```
ADDR OBJECT   LINE TAG     OPCD OPRNDS                  COMMENTS

1DB4 00
1DB5 6012     4947         DW   INP
1DB7 AB       4948         DB   171
1DB8 4C4F47   4949 LOGLI:  DB   'LOG',0
1DBB 00
1DBC A510     4950         DW   LOG
1DBE AB       4951         DB   171
1DBF 455850   4952 EXPLI:  DB   'EXP',0
1DC2 00
1DC3 EC10     4953         DW   EXP
1DC5 AB       4954         DB   171
1DC6 504F53   4955         DB   'POS',0
1DC9 00
1DCA 7612     4956         DW   POS
1DCC AB       4957         DB   171
1DCD 4C454E   4958         DB   'LEN',0
1DD0 00
1DD1 DF12     4959         DW   LEN#
1DD3 AB       4960         DB   171
1DD4 43485224 4961         DB   'CHR$',0
1DD8 00
1DD9 E512     4962         DW   CHR$
1DDB CB       4963         DB   203
1DDC 41534349 4964         DB   'ASCII',0
1DE0 49
1DE1 00
1DE2 F012     4965         DW   ASCII
1DE4 AB       4966         DB   171
1DE5 4E554D24 4967         DB   'NUM$',0
1DE9 00
1DEA FD12     4968         DW   NUM$
1DEC CB       4969         DB   203
1DED 56414C   4970         DB   'VAL',0
1DF0 00
1DF1 1013     4971         DW   VAL
1DF3 AB       4972         DB   171
1DF4 53504143 4973         DB   'SPACE$',0
1DF8 4524
1DFA 00
1DFB 3713     4974         DW   SPACE
1DFD CB       4975         DB   203
1DFE 53545249 4976         DB   'STRING$',0
1E02 4E4724
1E05 00
1E06 4713     4977         DW   STR$
1E08 D3       4978         DB   211
1E09 4C454654 4979         DB   'LEFT$',0
1E0D 24
1E0E 00
1E0F 5B13     4980         DW   LEFT$
1E11 D3       4981         DB   211
```

```
ADDR OBJECT   LINE TAG     OPCD OPRNDS             COMMENTS

1E12 52494748 4982         DB   'RIGHT$',0
1E16 5424
1E18 00
1E19 6413     4983         DW   RGHT$
1E1B D3       4984         DB   211
1E1C 4D494424 4985         DB   'MID$',0
1E20 00
1E21 6D13     4986         DW   MID$
1E23 DB       4987         DB   219
1E24 494E5354 4988         DB   'INSTR',0
1E28 52
1E29 00
1E2A A713     4989         DW   INSTR
1E2C BB       4990         DB   187
1E2D 00       4991         DB   0
1E2E 13       4992 PCHOF:  DB   19,20,0
1E2F 14
1E30 00
1E31 53544550 4993 STEPL:  DB   'STEP',0
1E35 00
1E36 5448454E 4994 THENL:  DB   'THEN',0
1E3A 00
1E3B 5049     4995 PILIT:  DB   'PI',0
1E3D 00
1E3E 01800000 4996 ONE:    DB   25165824           ;CONSTANT:   1
1E40   -      4997 NULLI:  EQU  *-2
1E42 02800000 4998 TWO:    DB   41943040           ;CONSTANT:   2
1E46 04A00000 4999 TEN:    DB   77594624           ;CONSTANT:   10
1E4A 02C90FD7 5000 PI:     DB   46731223           ;CONSTANT:   3.141593
1E4E 80FFFFFF 5001 NEGON:  DB   2164260863         ;CONSTANT:  -0.9999999
1E52 00B17216 5002 LN2:    DB   11629078           ;CONSTANT:   0.6931472
1E56 009714EB 5003 SQC1:   DB   9901291            ;CONSTANT:   0.59016206
1E5A 7FD5A956 5004 SQC2:   DB   2144708950         ;CONSTANT:   0.41730759
1E5E 01B504F3 5005 SQC3:   DB   28640499           ;CONSTANT:   1.41421356
1E62 00C90FD7 5006 QTRPI:  DB   13176791           ;CONSTANT:   0.7853892
1E66 00FFFFB8 5007 ATNCO:  DB   16777144           ;CONSTANT:   0.9999961
1E6A FFAA95BC 5008         DB   4289369532         ;CONSTANT:  -0.3331738
1E6E 7ECAD520 5009         DB   2127222048         ;CONSTANT:   0.1980737
1E72 FE8732D6 5010         DB   4270293718         ;CONSTANT:  -0.1323351
1E76 7DA3131C 5011         DB   2107839260         ;CONSTANT:   0.07962632
1E7A FC39A688 5012         DB   4236879544         ;CONSTANT:  -0.03360627
1E7E 79DF3A9E 5013         DB   2044672670         ;CONSTANT:   0.006812411
1E82 01C90FD7 5014 HALFP:  DB   29954007           ;CONSTANT:   1.570796
1E86 80A55DDE 5015 SINCO:  DB   2158321118         ;CONSTANT:  -0.64596371
1E8A 7DA33455 5016         DB   2107847765         ;CONSTANT:   0.079639679
1E8E F9993860 5017         DB   4187568224         ;CONSTANT:  -0.0046737656
1E92 749ED7B6 5018         DB   1956566966         ;CONSTANT 0.00015148419
1E96 00FFFEBC 5019 LOGCO:  DB   16776892           ;CONSTANT:   0.99998103
1E9A FFFFBAAC 5020         DB   4294949548         ;CONSTANT:  -0.4994712
1E9E 7FA80E29 5021         DB   2141720104         ;CONSTANT:   0.3282331
1EA2 FEE74B50 5022         DB   4276570960         ;CONSTANT:  -0.2258733
```

```
ADDR OBJECT   LINE TAG     OPCD OPRNDS             COMMENTS

1EA6 7E89DEE0 5023         DB   2122964704         ;CONSTANT:   0.1346393
1EAA FCE1C570 5024         DB   4242654576         ;CONSTANT:  -0.05511996
1EAE 7AB03FA8 5025         DB   2058370984         ;CONSTANT:   0.01075737
1EB2 7EFFFFA4 5026 EXPCO:  DB   2130706340         ;CONSTANT:   0.2499987
1EB6 7C8007F0 5027         DB   2088765424         ;CONSTANT:   0.03125758
1EBA 78A9D3F9 5028         DB   2024395769         ;CONSTANT:   0.002591371
1EBE 74B3E54E 5029         DB   1957946702         ;CONSTANT:   0.000171562
1EC2 6FB6350B 5030         DB   1874212107         ;CONSTANT:   0.0000054302
1EC6 6CB961A8 5031         DB   1824038483         ;CONSTANT:   0.0000006906
1ECA          5032 READY:  EQU  $
1ECA FD       5033         DB   253
1ECB 52454144 5034         DB   'READY',0
1ECF 59
1ED0 00
1ED1          5035 STOPM:  EQU  $
1ED1 FD       5036         DB   253
1ED2 53544F50 5037         DB   'STOP AT LINE ',254
1ED6 20415420
1EDA 4C494E45
1EDE 20
1EDF FE
1EE0 20455252 5038 ERRMS:  DB   ' ERROR IN LINE ',254
1EE4 4F522049
1EE8 4E204C49
1EEC 4E4520
1EEF FE
              5039 ;
              5040 ; VERB (STATEMENT/COMMAND) TABLE
              5041 ; FORMAT IS: DB 'VERB',0
              5042 ;            DW ADDR
              5043 ;            DB 'NEXT VERB',0
              5044 ;            ETC
              5045 ;  END OF TABLE IS MARKED BY DB 0
              5046 ;
1EF0          5047 JMPTB:  EQU  $
1EF0 4C495354 5048 LISTL:  DB   'LIST',0
1EF4 00
1EF5 5102     5049         DW   LIST
1EF7 52554E   5050 RUNLI:  DB   'RUN',0
1EFA 00
1EFB CB01     5051         DW   RUNCM
1EFD 4E4557   5052 NEWLI:  DB   'NEW',0
1F00 00
1F01 9E01     5053         DW   NEW
1F03 434F4E   5054 CONTL:  DB   'CON',0
1F06 00
1F07 DD02     5055         DW   CONTI
1F09 544150   5056 TAPEL:  DB   'TAP',0
1F0C 00
1F0D 3401     5057         DW   TAPE
1F0F 53415645 5058 SAVEL:  DB   'SAVE',0
```

```
ADDR OBJECT   LINE TAG     OPCD OPRNDS            COMMENTS

1F13 00
1F14 7401     5059         DW   SAVE
1F16 4B4559   5060 KEYLI:  DB   'KEY',0
1F19 00
1F1A 9101     5061         DW   KEY
1F1C 465245   5062 FREEL:  DB   'FRE',0
1F1F 00
1F20 AD01     5063         DW   FREE
1F22 4946     5064 IFLIT:  DB   'IF',0
1F24 00
1F25 0605     5065         DW   IF
1F27 52454144 5066 READL:  DB   'READ',0
1F2B 00
1F2C 6208     5067         DW   READ
1F2E 52455354 5068 RESTL:  DB   'RESTORE',0
1F32 4F5245
1F35 00
1F36 2303     5069         DW   RESTO
1F38 44415441 5070 DATAL:  DB   'DATA',0
1F3C 00
1F3D F401     5071         DW   RUN
1F3F 464F52   5072 FORLI:  DB   'FOR',0
1F42 00
1F43 0204     5073         DW   FOR
1F45 4E455854 5074 NEXTL:  DB   'NEXT',0
1F49 00
1F4A AF06     5075         DW   NEXT
1F4C 474F5355 5076 GOSBL:  DB   'GOSUB',0
1F50 42
1F51 00
1F52 4703     5077         DW   GOSUB
1F54 52455455 5078 RETLI:  DB   'RETURN',0
1F58 524E
1F5A 00
1F5B 2F03     5079         DW   RETUR
1F5D 494E5055 5080         DB   'INPUT',0
1F61 54
1F62 00
1F63 3E07     5081         DW   INPUT
1F65 5052494E 5082         DB   'PRINT',0
1F69 54
1F6A 00
1F6B 6503     5083         DW   PRINT
1F6D 474F     5084 GOTOL:  DB   'GO'
1F6F 544F     5085 TOLIT:  DB   'TO',0
1F71 00
1F72 E802     5086         DW   GOTO
1F74 4C4554   5087 LETLI:  DB   'LET',0
1F77 00
1F78 0E06     5088         DW   LET
1F7A 53544F50 5089 STOPL:  DB   'STOP',0
```

```
ADDR OBJECT    LINE TAG     OPCD OPRNDS                COMMENTS

1F7E 00
1F7F 0009      5090         DW    STOP
1F81 454E44    5091         DB    'END',0
1F84 00
1F85 F208      5092         DW    END
1F87 52454D    5093 REMLI:  DB    'REM',0
1F8A 00
1F8B F401      5094         DW    RUN
1F8D 21        5095         DB    '!',0
1F8E 00
1F8F F401      5096         DW    RUN
1F91 3F        5097         DB    '?',0
1F92 00
1F93 6503      5098         DW    PRINT
1F95 52414E44  5099 RANDL:  DB    'RANDOMIZE',0
1F99 4F4D495A
1F9D 45
1F9E 00
1F9F 2D09      5100         DW    RANDO
1FA1 4F4E      5101 ONLIT:  DB    'ON',0
1FA3 00
1FA4 4309      5102         DW    ON
1FA6 4F5554    5103 OUTLI:  DB    'OUT',0
1FA9 00
1FAA CA09      5104         DW    OUT
1FAC 44494D    5105 DIMLI:  DB    'DIM',0
1FAF 00
1FB0 420A      5106         DW    DIM
1FB2 4348414E  5107 CHALI:  DB    'CHANGE',0
1FB6 4745
1FB8 00
1FB9 BB09      5108         DW    CHANG
1FBB 444546    5109 DEFLI:  DB    'DEF'
1FBE 464E      5110 FNLIT:  DB    'FN',0
1FC0 00
1FC1 F401      5111         DW    RUN
1FC3 00        5112         DB    0                      ;END OF TABLE
1FC3           5113 ROMEN:  EQU   *-1
```

```
ADDR OBJECT   TAG      OPCD OPRNDS                 COMMENTS

1FC4                   ORG  $+4095/4096*4096       ;RAM STARTS ON 4K BOUNDARY
              ;
              ; ALL CODE ABOVE THIS POINT IS READ ONLY AND CAN BE PROM'ED
              ;
              ;
2000          RAM:     EQU  $
              ;
2000          LINEN:   DS   5
2005          IOBUF:   DS   10                     ;INPUT/OUTPUT BUFFER
200F          OUTA:    DS   3                      ;*** FILLED IN AT RUN TIME
2012          ADDR1:   DS   2                      ;HOLDS TEMP ADDRESS
2014          ADDR2:   DS   2                      ;HOLDS TEMP ADDRESS
2016          ADDR3:   DS   2                      ;HOLDS STMT ADDRESS DURING EXPR
2018          INDX:    DS   2                      ;HOLDS VARIABLE NAME OF FOR/NEXT
201A          REL:     DS   1                      ;HOLDS THE RELATION IN AN IF STM
201B          IFTYP:   DS   1                      ;HOLDS TYPE CODE OF LEFT SIDE
201C          TVAR1:   DS   4                      ;TEMP STORAGE
2020          TVAR2:   DS   4                      ;DITTO
2024          TEMP1:   DS   4                      ;TEMP STORAGE FOR FUNCTIONS
2028          TEMP2:   DS   4
202C          TEMP3:   DS   4
2030          TEMP4:   DS   4
2034          TEMP5:   DS   4
2038          TEMP6:   DS   4
203C          TEMP7:   DS   4
2040          LINEL:   DS   2                      ;HOLDS MIN LINE NUMBER IN LIST
2042          LINEH:   DS   2                      ;HOLDS MAX LINE NUMBER IN LIST
2044          PROMP:   DS   1                      ;HOLDS PROMPT CHAR
2045          LEN:     DS   1                      ;HOLDS LEN OF STMT
2046          EXPRS:   DS   2                      ;HOLDS ADDR OF EXPRESSION
2048          FACC:    DS   4
204C          FTEMP:   DS   12
2058          STRIN:   DS   256                    ;STRING BUFFER
2158          PARCT:   DS   1
2159          SPCTR:   DS   2
215B          CMACT:   DS   1                      ;COUNT OF COMMAS
215C          FNARG:   DS   4                      ;SYMBOLIC ARG & ADDRESS
0002          TTY:     EQU  2                      ;DEVICE ADDRESS FOR TERMINAL
2160          ILSW:    DS   1                      ;SWITCH, ON=INPUT LINE MODE
2161          STMT:    DS   2                      ;HOLDS ADDR OF CURRENT STATEMENT
2163          ENDLI:   DS   2                      ;HOLDS ADDR OF MULTI STMT PTR
2165          MULTI:   DS   1                      ;SWITCH 0=NO MULTI STMT LINE, 1=
2166          TAPES:   DS   1                      ;SWITCH 0=NOT INPUT FROM TAPE, 1
2167          RUNSW:   DS   1                      ;0=RUN MODE, 1=IMMEDIATE MODE
2168          DIMSW:   DS   1                      ;0=NORMAL EXPR, 1 =SUBSCRIPT EXP
2169          EDSW:    DS   1                      ;1=EDIT, 0=RUN
216A          DEXP:    DS   1
216B          COLUM:   DS   1                      ;CURRENT TTY COLUMN
216C          TRNDX:   DS   2
216E          TRNDY:   DS   2
2170          TRNDZ:   DS   2
```



```
ADDR OBJECT   TAG      OPCD OPRNDS                 COMMENTS

2172          TRNDS:   DS   2
2174          RNDX:    DS   2
2176          RNDY:    DS   2
2178          RNDZ:    DS   2
217A          RNDS:    DS   2
217C          RNDSW:   DS   1                      ;1=RANDOMIZE, 0=NO RANDOMIZE
217D          FNMOD:   DS   1                      ;SWITCH, 0=NOT IN DEF FN, <>0 =
217E          LINE:    DS   2                      ;HOLD ADD OF PREV LINE NUM
2180          STACK:   DS   2                      ;HOLDS ADDR OF START OF RETURN S
2182          FORNE:   DS   1                      ;NUMBER OF ENTRIES IN TABLE
2183                   DS   112                    ;ROOM FOR 8 NESTED FOR/NEXT
21F3          OUTSW:   DS   1                      ;SWITCH ON = OUTPUT, OFF = NO OU
21F4          PRSW:    DS   1                      ;ON=PRINT ENDED WITH , OR ; -- 0
21F5          NS:      DS   1                      ;HOLDS LAST TYPE (NUMERIC/STRING
21F6          DATAP:   DS   2                      ;ADDRESS OF CURRENT DATA STMT
21F8          DATAB:   DS   2                      ;ADDRESS OF DATA POOL
21FA          PROGE:   DS   2                      ;ADDRESS OF PROGRAM END
21FC          IMMED:   DS   73                     ;IMMEDIATE COMMAND STORAGE AREA
2245          BEGPR:   EQU  $
              ;
0000                   END  BASIC
```

```
                          *** C R O S S   R E F E R E N C E ***

TAG     DEFN ADDR  REFERENCES

ABS     2792 112A  2582 4920
ABSLI   4919 1D72
ADDR1   5125 2012  0423 0434 0452 0741 0776 0781 0921 0946 0950 0954 1151 1197
                   1213 1244 1261 1273 3329 3335 3734 3770 4752 4758 4890
ADDR2   5126 2014  3741 3749 3776 4837
ADDR3   5127 2016  3646 3705 3828 3860 4038 4055 4778 4781 4804 4845 4883
ALPHA   4491 1B2C  3407 3442 4635
ARG     4748 1C63  3745 4765
ARGNU   4763 1C72  3229 3244 3255 3267 3270 3312
ASCII   3142 12F0  4965
ATBEG   4144 1972
ATN     2690 105E  4944
ATN1    2701 1060  2696 2698
ATNCO   5007 1E66  2721
ATNLI   4943 1DAA
BASIC   0046 0001  5186
BEGPR   5184 2245  0157 0222 0354 0400 0487 0564 0595 0632 1568 4808
BINFL   4893 1D47  0376 3037
BSCHA   4133 195C
CCRUN   4274 1A26
CHA1    1601 09E4  1614
CHA2    1618 09F9  1580
CHA3    1647 0A29  1660
CHALI   5107 1F82
CHANG   1571 09BB  5108
CHR$    3129 12E5  4962
CMACT   5150 215B  3957
COLUM   5162 216B  0706 3045 4124 4183 4185 4199 4209
COMP1   0073 0013  4317
COMP2   4298 1A4D  0075
COMP3   4304 1A51  4312
COMP4   4305 1A52  4313
COMP5   4308 1A55  4303
COMP6   4314 1A62  4309
CONC2   3069 1292  3064
CONC3   3073 129A  3078
CONC4   3079 12A2  3072
CONC5   3092 12B2  3094
CONC6   3107 12CC  3112
CONCA   3048 127C  3908
CONTC   4270 1A1C
CONTI   0540 02DD  5055
CONTL   5054 1F03
CONTO   4280 1A35
COPYD   4687 1C24  0103 0162 0165 0399 0886 1505 2257 2459 2718 4696 4704
COPYH   4699 1C2D  0094 0248 0283 2201 2266 2331 2369 2783 3017
COS     2650 1031  2672 4938
COSLI   4937 1D9C
CR1     4096 1911  4103
```

```
                    *** C R O S S   R E F E R E N C E ***

TAG     DEFN ADDR  REFERENCES

CRLF    4116 1932  0657 4095 4129 4177 4180 4201
CVERR   4375 1AB0  1260 1335 1419 3202
DAERR   4371 1AA8  1387
DATAB   5131 21F8  0146 0358 0367 0394 4529
DATAL   5070 1F38  1390
DATAP   5180 21F6  0401 0596 1379 1385 1393 1407 1428
DEFLI   5109 1FBB  4315
DEXP    5161 216A  1990 1996 2067 2880 2905
DIM     1665 0A42  1741 5106
DIM1    1687 0A60  1691
DIM2    1729 0A92  1734
DIM3    1736 0A9D  1771
DIM4    1742 0AA8  1739
DIMLI   5105 1FAC
DIMSW   5159 2168  0154 1359 1676 3829 3838 3857 4010 4030 4034 4037 4789 4792
                   4374
ECHO    4152 1985  4139 4149
EDIT1   0212 00D5  0218
EDIT2   0273 00E6  0254
EDIT3   0244 0106  0249
EDIT4   0255 011A  0230 0235 0246
EDIT4   0255 011A  0230 0235 0246
EDIT5   0258 011F  0226 0231 0236
EDIT6   0269 0134  0276 0279
EDIT7   0277 0140  0275
EDM1    3978 1840  3840
EDM2A   3985 184F  3998
EDM3    3988 1853  3981
EDM4    3999 1867  3859
EDSW    5160 2169  0196 0390 0557 1015 4271
END     1456 08F2  0422 0491 0513 0520 5092
ENDLI   5155 2163  0410 0611 0626 0826 4333
ENDXP   3643 15E0  3692
EOL     4319 1A68  0563 0594 0604 0623 0718 0769 1023 1120 1191 1270 1289 1451
                   1473 1499 1564 1595 1632 1742 4864
EOL0    4331 1A79  4323
EOL1    4332 1A7C  4330
ERRMS   5033 1EE0  4682
ERROR   4672 1C0E  0085
EV0     3707 1661
EV1     3708 1664  3718
EV10    3881 179D  3864 3867 3870 3873
EV11    3931 17F5  3902
EV1A    3713 166C  3731
EV2     3719 1674  3712
EV2A    3733 1694  3726 3728
EV3     3748 16AB  3742 4891
EV3A    3769 16CC  3823
EV4     3796 16F0  3755
```

*** C R O S S   R E F E R E N C E ***

```
TAG     DEFN ADDR  REFERENCES

EV4A    3807 1704  3809
EV4B    3816 170F  3821
EV5     3824 171C  3732
EV6     3836 173A  3825
EV7     3851 1756  3856
EV8     3857 1760  3853
EV9     3862 176B  3837
EVAL    3706 165E  3113 3647 3795 3835 3940 3959
EVCOM   3955 181A  3875
EVLD    3960 1826  3849 3839 3948 3995 4760
EVLD1   3970 1837  3975
EVNEG   3941 1806  3877
EXEC    0236 0150  0203
EXEC1   0301 0160  0306
EXP     2760 10EC  3925 4953
EXP1    2777 1106  2767 2769
EXPCO   5036 1EB2  2785
EXPLI   4952 1DBF
EXPR    3373 1407  0671 0703 0734 0747 0764 0875 0920 1022 1361 1436 1450 1515
                   1677 4863
EXPRS   5144 2046  3389 3399 3401 3431 3433 3500 3511 3532 3539 3545 3548 3592
                   3599 3616 3619 3650 3676 3687 3694 3697 3706 4025 4029 4040
                   4445 4799 4866
FACC    5145 2048  0092 0101 1168 1846 1864 1870 1872 1886 1914 1922 1931 1949
                   1955 1971 1985 1992 2008 2019 2025 2032 2052 2053 2179 2210
                   2229 2234 2239 2258 2263 2267 2281 2351 2373 2427 2473 2511
                   2514 2516 2602 2673 2734 2750 2787 2789 2798 2800 2820 2833
                   2844 2853 2864 2882 2907 2922 2934 2944 2947 4723 4736 4898
FAD1    2547 0F96  2553
FADD    2166 0D59  1155 2336 2594 2631 2658 2702 2843 2891 2920 3909
FADD3   2191 0D7C  2188
FADD4   2197 0D89  2187 2192
FADD6   2219 0DAE  2209
FADD7   2221 0DB0  2227
FADD9   2223 0DBC  2204
FADDA   2254 0DEF  2233
FADDB   2258 0DF7  2272
FADDJ   3907 17BF  3863
FADDT   2542 0F95  1952 1959 2028 2036 2237 2397 2469
FALSE   0961 01F4  0942 0944 0945 0996
FBIN    4718 1C3B  0705 1441 1452 1516 1642 1655 3027 3135 3211 3227 3999 4005
                   4766
FBIN0   4738 1C58  4741
FBIN1   4742 1C5E  4737
FBIN2   4744 1C60  4727 4731
FDEC    3035 1270  1604 2739 2743 3046 3127 3151 3154 3186 3342
FDIV    2411 0EE3  1906 2002 2611 2687 2708 2775 2918 3872
FDIV3   2446 0F12  2453
FDIV5   2461 0F28  2490
```

*** C R O S S   R E F E R E N C E ***

```
TAG     DEFN ADDR  REFERENCES

FDIV6   2471 0F42  2465
FDIV7   2474 0F48  2479
FDIV8   2483 0F56  2488
FEXP    2519 0F83  1986 1993 2009 2181 2184 2244 2303 2353 2356 2425 2429 2603
                   2735 4729
FIN     1835 0B13  1255 1402 3199 3396
FIN2    1854 0B31  1852
FIN3    1878 0B5D  1874
FIN4    1886 0B67  1858
FIN5    1893 0B73  1903
FIN6    1901 0B81  1907
FIN7    1904 0B86  1897
FIN8    1908 0B8E  1849 1869
FIN9    1915 0B9E  1853 1910 1927 1942
FINA    1917 0BA0  1912
FINB    1923 0BB6  1924
FINC    1943 0BCE  1928
FIND    1955 0BE9  1944 1947 1948
FMUL    2338 0E6C  1900 2000 2615 2619 2629 2643 2712 2745 2788 2790 2889 2910
                   3869 3924 3930
FMUL1   2351 0E77
FMUL5   2375 0EA6  2382
FMUL6   2384 0EB2  2403
FMUL7   2387 0EB8  2392
FMUL8   2398 0ECE  2393
FMUL9   2401 0ED4  2406
FN      4768 1C78  3720 3722
FN2     4809 1CBA  4831
FN3     4827 1CD8  4817 4823
FN4     4832 1CE0  4826
FNARG   5151 215C  4585 4793 4795 4839 4870 4872
FNL     3495 14D7  3456
FNL2    3499 14DD
FNL3    3511 14EF  3509
FNLIT   5110 1FBE  3454
FNMOD   5172 217D  0296 4526 4786 4803 4877
FNORM   2274 0E1D  1890 2262 2409 2491 2869 3013 3018 4906
FNRM1   2291 0E2B  2287 2316
FNRM2   2298 0E35  2303
FOR     0721 0402  5073
FOR1    0763 0447  0760
FOR2    0765 044B  0756 0762
FOR3    0776 0460
FOR4    0781 046F  0775
FOR5    0785 047B  0773 0779 0783
FOR6    0794 048B  0806
FOR7    0802 0497  0796
FOR8    0807 049E  0793 0801
FOR9    0839 04CF  0780 0784 0852 0856 0862 0866
```

*** C R O S S   R E F E R E N C E ***

```
TAG    DEFN ADDR  REFERENCES

FORA   0863 04FD  0860
FORLI  5072 1F3F
FORNE  5175 2182  0152 0392 0785 0810 1124 1143 1146 1172
FOUT   1961 0BF3  0378 0678 3166
FOUT0  1979 0C04  1977
FOUT1  1983 0C0B
FOUT2  1985 0C0E  1980
FOUT3  1992 0C1F  2001 2003
FOUT4  1996 0C2A  2005 2016
FOUT5  2002 0C3A  1999
FOUT6  2004 0C40  1995
FOUT7  2018 0C63  2030
FOUT8  2032 0C88  2020 2023 2024
FOUT9  2037 0C92  2017 2029 2088 2090
FOUTA  2047 0CA1  2044
FOUTB  2050 0CA4  2011
FOUTC  2053 0CA9  2062
FOUTD  2055 0CAE  2060
FOUTH  2064 0CBB  2031
FOUTI  2079 0CD2  2071
FOUTJ  2082 0CD6  2084
FOUTK  2099 0CF5  2095
FOUTL  2101 0CF8  2098
FOUTM  2103 0CFB  2106
FOUTN  2118 0D13  2126
FOUTO  2127 0D1E  2116 2147 2164
FOUTP  2128 0D1F  2133
FOUTQ  2134 0D2B  2130
FOUTR  2140 0D33  2117
FOUTS  2148 0D41  2141
FOUTT  2156 0D4B  2162
FOUTU  2017 0C60  2013
FOUTV  1989 0C19  1987
FOV1   2502 0F6F  2498
FOVUN  2493 0F65  2358 2432
FREE   0362 01AD  5063
FREEL  5062 1F1C
FSB1   2533 0F8B  2539
FSHFT  2556 0FA0  2224 2252 2565
FSUB   2318 0E53  0777 0732 0929 1162 2587 2591 2706 2752 3866 3918
FSUBT  2528 0F8A  2261 2464
FTEMP  5146 204C  1854 1860 1878 1925 1945 1950 2006 2014 2021 2026 2199 2205
                  2230 2235 2255 2259 2264 2268 2329 2332 2366 2370 2385 2394
                  2375 2399 2442 2461 2462 2466 2467 2482
FTEST  2506 0F75  0770 0933 0938 0943 1975 2175 2177 2195 2238 2253 2293 2327
                  2344 2417 2579 2695 2730 2766 2812 2877 2931
FUNC0  3462 14A2  3518
FUNC1  3464 14A6  3476
FUNC2  3470 14B0
```

```
                        *** C R O S S   R E F E R E N C E ***

TAG     DEFN ADDR  REFERENCES

FUNC3   3477 14BB  3469
FUNC4   3490 14CF  3467
FUNCT   4918 1D72  3463
GETCH   4266 1A13  4087
GETCM   0186 0086  0178 0260 0284 0335 0360 0381 0419 1020
GETS1   1319 080E  1323
GETS2   1329 081F  1316 1318
GETS3   1330 0820  1339
GETS4   1340 0830  1333
GETS5   1342 0832  1320 1322
GETS8   1352 0840  1009 1241 1376
GETS9   1368 085D  1356
GETST   1312 07FE  1272 1420
GOSBL   5076 1F4C  0951 1523
GOSU1   0624 0350  1567
GOSUB   0616 0347  0953 5077
GOTO    0551 02E3  0548 0949 0959 5086
GOTO1   0561 02EF
GOTO2   0565 02F3  0537 0633 1569
GOTO3   0583 031B  0571 0576
GOTOL   5084 1F6D  0947 1523
GTEMP   4458 1B08  3400 3935 3952
HALFP   5014 1E32  2586 2590 2593 2610 2657
HDRI    4385 1AB6  4389
HDRTL   4378 1AB4  0322 1465
IF      0869 0506  5065
IF1     0883 0524  0880
IF2     0890 0528  0887
IF3     0894 052C  0919
IF4     0900 0535  0897
IF5     0905 053D  0901
IF6     0911 0547  0906
IF7     0921 055A
IF8     0935 057F  0932
IF9     0940 058D  0937
IFB     0958 05BC
IFF     0962 05C5  0927
IFG     0967 05CE  0992
IFH     0973 05D7  0971
IFI     0978 05DF  0975
IFJ     0986 05ED  0984
IFK     0990 05F3  0988
IFL     0994 05FA  0999 1001
IFLIT   5064 1F22
IFM     0998 0604  0980
IFN     1000 0609  0981
IFTYP   5130 201B  0878 0923
ILSW    5153 2160  0156 1291 1305 4097
IMMED   5133 21FC  0205 0220 0258 0281 0300 0309
```

*** C R O S S   R E F E R E N C E ***

TAG	DEFN	ADDR	REFERENCES
INDX	5128	2018	0729 0736 0786 0854 0857 1122
INIT1	0130	0049	0049
INIT2	0134	0052	0142
INIT3	0143	005F	0138
INP	3020	1260	4947
INPL	1276	07C2	1210
INPL1	1297	07E4	1304
INPL2	1305	07F0	1299
INPLI	4946	1DB1	
INPU1	1216	0750	1268
INPU2	1221	075B	1218
INPU3	1223	075F	1229
INPU4	1230	076A	1226
INPU5	1240	077D	1237
INPU6	1241	077E	1220 1239
INPU7	1251	0792	1248
INPU8	1261	07A5	1258
INPU9	1264	07AA	1274
INPUA	1272	07B9	1254
INPUB	1270	07B3	1416
INPUT	1201	073F	5081
INS	3679	162A	3669 3672 3674 3686
INS0	3665	160D	3658
INS1	3671	1613	3657
INS3	3670	1617	3666
INST1	3316	1381	3320
INST2	3318	1386	3315
INST3	3321	138D	3317 3319
INST4	3328	13C5	
INST5	3341	13D9	3362 3371
INST6	3343	13DD	3339 3369
INST7	3347	13E3	
INST8	3352	13E8	3358
INST9	3363	13F9	3355
INSTA	3370	1402	3346
INSTR	3303	13A7	4989
INT	2823	1145	4726 4926
INT2	2839	1153	2836
INT3	2846	1162	2841
INT4	2853	116C	2863
INT5	2856	1172	2861
INTLI	4925	1D80	
IOBUF	5123	2005	0082 0299 0377 0521 0643 0881 0962 1212 1222 1296 4071 4106 4171 4680
JMPTB	5047	1EF0	0430
KEY	0337	0191	1466 4273 4277 5061
KEYLI	5060	1F16	
LDALP	3405	1436	3395
LDBTN	3542	1526	3408

*** C R O S S   R E F E R E N C E ***

TAG	DEFN	ADDR	REFERENCES
LDDTP	3552	153A	3543
LDF	3397	1426	3523
LDFN	3442	147B	3413
LDFN1	3444	1481	3447
LDFNC	3530	1514	3489
LDNUM	3396	1423	3406
LDPI	3524	150A	3461
LDRND	3517	14FC	3451
LDV	3431	1465	3529 4057
LDV1	3416	144D	3443 3494
LDV2	3425	145A	3419
LDV2A	4012	187E	3426
LDV2B	3427	145F	
LEFT$	3238	135B	4980
LEN	5143	2045	
LFN#	3120	12DF	4959
LET	1003	060E	0453 0549 5088
LET1	1021	0629	1014
LET2	1034	0643	1029
LET2A	1039	064E	1036 1309 1349 1662
LET3	1047	065A	1049
LET4	1059	0668	1077
LET5	1069	0674	1076
LET6	1078	0681	1061
LET7	1090	0696	1087 1088
LET8	1102	06A3	1042
LET9	1105	06A6	1110
LETLI	5087	1F74	
LINE	5173	217E	0308 0424 1194 1486 4665
LINEH	5141	2042	0467 0476 0486 0508
LINEL	5140	2040	0465 0475 0496
LINEN	5122	2000	0545 1487 4664 4669
LINEO	4218	19DD	0522 1488 4666
LIST	0455	0251	0323 5049
LIST1	0487	0281	0470 0480
LIST2	0488	0284	0532 0533
LIST3	0492	028C	
LIST4	0507	02A4	0501
LIST5	0515	02B4	0512
LIST6	0521	02BE	0514 0519
LIST7	0523	02C4	0528
LIST8	0533	02D4	0500 0506
LISTL	5048	1EF0	
LIT1	3563	1554	3571
LIT2	3586	1578	3588
LIT3	3605	1594	3610
LIT4	3611	159C	
LITST	3560	154E	3557
LLINE	4916	1D69	1207

*** C R O S S   R E F E R E N C E ***

```
TAG     DEFN ADDR  REFERENCES

LN2     5002 1E32  2744
LOG     2724 10A5  3922 4950
LOG1    2742 10C4  2736
LOG2    2744 10C9  2741
LOGCO   5019 1E96  2757
LOGLI   4949 1DB8
LOOKD   3392 141C  3516 3541 3551 4035
LOOKO   3622 15B2  3404 3437 3441 3612 3831 3834
LOUT    4231 19E8  4226 4227
MID$    3260 136D  4986
MID0    3273 1377  3247 3258
MID1    3280 1385  3276 3277
MID2    3286 1392  3231 3283
MID3    3290 1396  3285
MID4    3295 139E  3300
MULTI   5156 2165  0295 0405 0581 0613 0624 1480 4331
NEG     2925 110A  2596 2697 2699 2740 2768 3949
NEGON   5001 1E4E  2842
NEW     0348 019E  5053
NEWLI   5052 1EFD
NEXT    1113 06AF  5075
NEXT1   1131 06C5  1141
NEXT2   1138 06D0  1134
NEXT3   1143 06D9  1137
NEXT4   1170 0708
NEXT5   1176 0714  1169
NEXT6   1178 0718  1163 1171
NEXT7   1172 070C  1177
NEXTL   5074 1F45  0850
NOTBS   4147 197A  4132
NOTC    4151 1982
NOTCO   4131 1957  4127
NOTCR   4126 1949  4092
NOTEQ   4350 1A8C  0561 0621 1551
NOTZ1   4241 19F6  4238
NOTZ2   4250 1A06  4247
NRNDS   4914 1D62
NRNDX   4911 1D5C  0159 0164 0396
NRNDY   4912 1D5E
NRNDZ   4913 1D60
NS      5179 21F5  0675 0877 0922 1028 3897 3962
NULLI   4997 1E40  0307
NUM$    3156 12FD  4968
NUM1    3169 1308  3172
NUMEN   4515 1B3B  4502
NUMER   4504 1B35  0958 1545 1556 3394 3412 4400 4641
NXERR   4373 1AAC  0845 1123 1142
OBRCE   3613 15A0  3555
ON      1508 0943  5102
```

*** C R O S S   R E F E R E N C E ***

```
TAG     DEFN ADDR  REFERENCES

ON3     1534 0969  1576
ON3A    1535 096A  1544 1550
ON4     1545 097D  1542 1548
ON5     1549 0987  1546
ON6     1551 098B  1537 1555
ON7     1556 0997  1553
ON8     1559 09A0  1563
ON9     1564 09A8  1561
ONE     1996 1E3E  0751 2701 2751 2772 2780 2817
ONLIT   5101 1FA1
OP1     3648 15ED  3625 3628 3631 3634 3641
OP2     3690 163B  3633
OP3     3700 1650  3636
OPLP1   3654 15F8  3664
OPLP2   3660 1604  3663
OPLP3   3659 1602
OT1     4172 1991  4187 4188
OT2     4179 19A0  4176
OT4     4186 19AF  4178
OUT     1431 08CA  5104
OUTA    5124 200F  1442 1453 3028 3034
OUTLI   5103 1FA6
OUTSW   5177 21F3  0155 0655 4168 4196 4231 4283
OVERR   4363 1A98  2308 2423 2501 2504 2731 2879
PACK    4391 1ACC  0200 0471 0483 0562 0622 1558
PARCT   5148 2158  3382 3614 3643 3690 3700 3702 3832 4017 4020 4044 4734 4879
PCHOF   4992 1E2E  0344
PI      5000 1E4A  3528
PILIT   4995 1E3B  3459
PK1     4399 1AC8  4421
PK3     4409 1ADC  4416
POS     3039 1276  4956
POWER   3910 17C7  3379
PRCNT   4286 1A41  4128 4270 4280
PRIN1   0643 0369  0701
PRIN2   0646 0370  0682 0696 0710 0713
PRIN3   0652 037B  0719
PRIN4   0659 038C  0645
PRIN5   0683 03BD  0677
PRIN6   0688 03C6  0693
PRIN7   0694 03CE  0686
PRIN8   0697 03D3  0662
PRIN9   0698 03D6  0664
PRINA   0702 03DF  0669
PRINB   0713 03F3  0717
PRINC   0718 03FC  0660
PRINT   0635 0365  0651 5083 5098
PRLIN   4660 1BFC  1476 4276 4684
PROGE   5182 21FA  0257 0263 0267 0357 0369 3386 4797 4801 4868
```

*** C R O S S   R E F E R E N C E ***

```
TAG     DEFN ADDR  REFERENCES

PROMP   5142 2044  4070 4077
PRSW    5178 21F4  0642 0652 0700 4398 4404 4407
QTRPI   5006 1E62  2715
RAM     5120 2000  0047
RANDL   5099 1F95
RANDO   1493 092D  5100
RDY     0168 0098  0346 1464 1479 1491 4685
RDY1    0176 009E  0183
RDY2    0184 00B3  0181
READ    1372 0862  1414 5067
READ1   1384 0870  1397
READ2   1398 088B  1382 1392
READ3   1407 0898  1418
READ4   1410 08A0  1429
READ5   1417 08AC  1405 1425
READ6   1420 08B3  1401
READ7   1427 08C3  1423
READ8   1428 08C4  1426
READL   5066 1F27
READY   5032 1ECA  0174
REDIM   1744 0AAE  1697
REDM1   1760 0AC4  1755
RFIN    4071 18D8  4105 4130
REL     5129 201A  0916 0930 0935 0940 0994 1522 1532 1565
REMLI   5093 1F87
RESTL   5068 1F2E
RESTO   0589 0323  5069
RETLI   5073 1F34
RETUR   0599 032F  5079
RGHT$   3749 1364  4993
RND     2937 11E4  0182 3520 4932
RND1    2949 11F2  3009
RND2    2952 11F8  2957
RND3    2961 1206  2985
RND4    2970 1215  2965
RND5    2982 1221  2966 2969
RND6    2987 1229  2999
RNDLI   4931 1D8E  3449
RNDP    4908 1D56  2959
RNDQ    4909 1D58
RNDR    4910 1D5A
RNDS    5170 217A  2987 3004
RNDSW   5171 217C  0176 0184 0393 1501 3011
RNDX    5167 2174  0397 1503 3015
RNDY    5168 2176
RNDZ    5169 2178  2949
ROMEN   5113 1FC3
RST1    0052 0008  0061
RST2    0064 0010
```

*** C R O S S   R E F E R E N C E ***

```
TAG     DEFN ADDR  REFERENCES

RST3    0078 0018
RST5    0088 0020  2178 2196 2350 3965
RST6    0097 0030
RST7    0106 0038
RST7A   0170 0046  0118
R1ERR   4369 1AA4  0607
RUN     0405 01F4  0597 0614 0654 0658 0838 0867 0961 1033 1037 1175 1199 1271
                   1310 1454 1506 1615 1663 1743 5071 5094 5096 5111
RUN1    0412 0204  0408
RUN2    0417 020E  0404 0582
RUN3    0427 0220  0310 0411
RUN4    0428 0221  0960
RUN5    0431 022A  0451
RUN6    0447 0242  0436
RUN7    0452 024B  0433
RUNCM   0383 01CB  5051
RUNLI   5050 1EF7
RUNSW   5158 2167  0298 0391 0417 0558 1477
SADR    4483 1B24  3432
SAVE    0312 0174  5059
SAVEL   5058 1F0F
SCH0    4529 1B47  4588 4592
SCH1    4530 1B4A  4557
SCH2    4556 1B6C  4542 4547
SCH3    4558 1B70  4532
SCH4    4573 1B88  4569
SCH5    4575 1B8C  4572 4578
SCH6    4585 1B9A  4528
SCH7    4599 1BAE  4564
SCH8    4612 1BC3  4617
SCH9    4621 1BCF  4550
SEARC   4520 1B3F  0739 1149 1297 1357 1368 1581 1590 1621 1633 1672 3430 4049
SGN     2803 1133  4929
SGNLI   4928 1D37
SIN     2568 0FA9  2659 2685 4935
SIN1    2583 0FB7  2580 2592
SIN2    2586 0FBB  2581
SIN3    2593 0FD0  2588 2589
SIN3A   2610 0FEA  2604
SIN4    2627 100D  2722 2758 2786
SIN5    2629 100F  2648
SINCO   5015 1E86  2623
SINLI   4934 1D95
SKP2Z   4448 1B04  0438 0447 3470 3477 4456
SKPP    3550 1536  3553 3621 3689 3699
SNERR   4367 1AA0  0732 0746 0809 0914 0925 0957 1017 1031 1035 1286 1439 1518
                   1530 1540 1557 1585 1627 1631 1671 1756 1759 1789 1793 1813
                   1817 3486 3515 3559 3568 3645 3703 3899 3989 4326 4358 4406
                   4636 4811 4835 4857 4861
```

*** C R O S S   R E F E R E N C E ***

```
TAG     DEFN ADDR  REFERENCES

SPACI   3215 133F  3219
SPACE   3205 1337  4974
SPCTR   5149 2159  3085 3089 3335 3531 3533 3757 3760 3800 3804 3850 4021 4024
                   4042 4477 4480 4732 4831
SQC1    5003 1E56  2388
SQC2    5004 1E5A  2390
SQC3    5005 1E5E  2909
SQR     2871 118A  4923
SQR1    2915 11C4  2898 2901
SQRLI   4922 1D79
SQUI2   4439 1AF3  4444
SQUIS   4423 1AED  3117 3794 3827 3939 3956
STACK   5174 2180  0151 0197
STEPL   4993 1E31  0758
STERR   4365 1A9C  3843 3880
STMT    5154 2161  0403 0412 0416 0579 0609 0628 0833 0836 0839 0846 1136 1139
                   1192 1481
STOP    1468 0900  5090
STOPL   5039 1F7A
STOPM   5035 1ED1  1474 4274
STR$    3221 1347  4977
STRI1   3232 1354  3235
STRIN   5147 2058  0683 0882 0964 1039 1083 1295 1307 1312 1342 1636 3055 3069
                   3081 3103 3126 3136 3148 3163 3182 3187 3198 3212 3223 3230
                   3274 3294 3313 3561 3573 3601 3796 3966
SUB1    1794 0AE5  1790
SUB2    1803 0AF0  1308
SUB3    1809 0AFA  1305
SUB4    1813 0B09  1314
SUBSC   1773 0AD2  1363 4053
TABLI   4917 1D6E  0667
TABST   4190 1983  0697
TAN     2661 103A  4941
TAN1    2676 1049  2684
TANLI   4940 1DA3
TAPE    0325 0184  5057
TAPEL   5056 1F09
TAPES   5157 2166  0153 0319 0332 0343 1462 4074 4093 4133 4147
TBLP    4204 19C2  4208
TBLP2   4207 19C9  4205
TBON    4209 19CE  4206
TBSPA   4212 19D3  4216
TEMP1   5133 2024  2600 2630 2640 2703 2707 2716 2746 2770 2774 2781 2892 2899
                   2917 2919 3911 3920 3927 3929 3983 3996 4003
TEMP2   5134 2028  2612 2614 2618 2620 2644 2709 2711 2732 2753 2777 2883 2915
TEMP3   5135 202C  2616 2713
TEMP4   5136 2030
TEMP5   5137 2034
TEMP6   5138 2038
```

*** C R O S S   R E F E R E N C E ***

```
TAG    DEFN ADDR  REFERENCES

TEMP7  5139 203C  2670 2674 2686 3915 3923
TEN    4999 1E46  1896 1998
TERMI  4059 18D5  0199 1250 1293
TERMM  4188 1991  0175 0345 1019 1475 4275 4670 4681 4683
TERMO  4155 1989  0330 0530 0649 1233
TESTO  4108 1927  0321 0334 4078 4082 4117 4119 4121 4122 4141 4143 4145 4151
                  4182 4213 4293 4296 4386
THENL  4994 1E36  0955
TOLIT  5085 1F6F  0744 1583 1625
TOUT1  4110 1928  4112
TREAD  4083 18F5  4076 4080 4086 4090 4135 4146 4153
TRNDS  5166 2172
TRNDX  5163 216C  0160 1502 3014
TRNDY  5164 216E
TRNDZ  5165 2170
TRUE   0946 059E  0934 0939 0997
TSTC1  4278 1A32  4269
TSTCC  4257 1A0E  0429 0488
TSTEL  4336 1A81  0659 1560 4325 4357
TTY    5152 0002  0130 0132 0179 4084 4110 4114 4263 4266
TVAR1  5131 201C  0749 0772 0821 0888 0928 1156 1195
TVAR2  5137 2020  0766 0816
TWO    4998 1E42  2705 3917
ULERR  4361 1A94  0567 0577
VAL    3176 1310  4971
VAL1   3188 131C  3195
VAL2   3194 1325  3191
VAR    4627 1BD4  0727 1119 1283 1352 1577 1586 1628 1670 3496 4819 4837
VAR2   4646 1BE9  4642
VAR3   4653 1BF4  4647
VAR4   4655 1BF7
WHATL  4915 1D64  1018
XSQR   3927 17EB  3919
Z1     4245 19FC  4240
Z2     4254 1A0C  4249
ZEROM  4708 1C33  1848 1866 2372 2838 2945 3984 4715
```

IMSAI 8 K BASIC

TTY LISTING OF TAPE

~~AA&IIII2\~~

~~\AA6IIII6"AIII6\~~

\&I~~III2AA>AAA" @ IIIA>F&LIII2AA>AAA>00"AII6>QIE>~~

@B

```
>III2
:1E000000210024 3EFAC349007EFE20C023C30800C50600CF1AC34D1A2105207023C395
:1E001E000E1C000000000000000000001148200604C32D1C1148200604C3241CE37E23E31E
:1E003C00D55FB71600F2460016FF19D1C9D3033E17D3030100047E2F77BEC25F002F6D
:1E005A007709D25200F90100FF0922F821AFF52100003922802132822132662132 68BE
:1E0078002132F32132602121442277115C1D216C210608CD241C0608115C1DCD241C5A
:1E009600360321CA1ECD91193A7C21FE03C2B600DB03E602C2B300CDE411C39E0032B3
:1E00B4007C213E3A3269212A8021F9CDD518CDC01A78B1CA5001C511FD21EB70237111
:1E00D20023060 31A77042313B7C2D5007832FC21C1214522 7EE5B7CA1F01237EB8DA84
:1E00F0001201C21F01237EB9DA1201C21F012B2B545D06004E097EB7CA1A0147CD2D15
:1E010E001CC30601E116005E19C3E600EB7722FA213AFC21FE04CAB6004F06002AFAE5
:1E012C0021545D0922FA21C11A771B2B7AB8CA4001D234017BB9D23401 1321FC2146EF
:1E014A00CD2D1CC3B600326521327D213C3267212105201 1FC217E121323B7C2600176
:1E0168002140 1E227E2121FC21C320023E023266213E12CD2719CDB41AC351023E01D0
:1E018600326621 3E11CD2719C3B600AF32662121 2E1ECD9119C398002145 22AF772256
:1E01A400FA212AF82177C3B6002AF821EB2AFA217B955F7A9C57CD471D210520CDF369
:1E01C2000B3600CD8919C3B600AF3269213267213282213 27C212AF821771 15C1D21C8
:1E01E00074210608CD241C21442222F62123226121C30E022165217EB7CA0402360015
:1E01FE002A6321C320022A61215E1600192261213A6721B7C2B6007EB7CAF208232244
:1E021C007E212323CF224220CD0E1A11F01E1AB7CA4B022A4220D7C24202E5CD041B56
:1E023A00131A6F131A67E3C9CD041B131313C32A022A4220C30E06CF110000EB223A2C
:1E025800202199992 23C20EBB7CA8102CDC01A5059EB223A20223C20EBCFFE2CC2814C
:1E0276000223CFCDC01A6069223C2021 4522CD0E1A7EB7CAF208D6034723EB2A3A2060
:1E029400EB7EBADAD402C2A402237E2BBBDAD402EB2A3C20EB7EBACAB402D2F208C337
:1E02B200BE02237E2BBBCABE02D2F208110520CDDD197E121323 05C2C402E5CD8919F1
:1E02D000E1C3840258160019232 3C384022100207EB7C2E802C30E06AF3269213267D3
:1E02EE0021CD8C1ACDC01ACD681A2145227EB7CA941A237EB8DA1B03237E2BB9DA1B68
:1E030C0003C2941A2B226121AF326521C30E022B5E160019C3FB02CD681A21442222E7
:1E032A00F621C3F401CD681AF1FEFFC2A41AE1226121E1226321F1326521C3F401CDEF
:1E0348008C1ACDC01ACD681A3A6521F52A6321E52A6121E53EFFF5214522C3FB02AFF9
:1E036600 32F421110520CFC38C033EFE12E5CD8919E1C365033AF421B7C2F4013AF343
:1E03840021B7CC3219C3F401CD811ACAFC03FE2CCAD303FE3BCAD603D5E5116E1DD7B0
:1E03A200CADF03E1CD0714D1E5EB3AF521FEE7CABD03CDF30B23EBE1C370031 15820EF
:1E03C0001AB7CACE0347131A772305C2C603EBE1C37003CDB319233E0132F421C369A5
:1E03DE0003D1CD0714E5CD3B1C216B2196E1D1FA7003473E20CA7003121305C3F30315
:1E03FC00CD681AC37B03CDD41BEB221220EBFE3DC2A01A23CD0714EB2A1220EBE5CDC7
:1E041A003F1BEB224220F7E1116F1FD7C2A01ACD0714E5211620F7213E1EEFE17EB794
:1E043800CA4B04E511311ED7CA4704E1C34B04D1CD0714E5211A20F7E1CD681ACD7507
:1E045600 0FF5211620EFF1F26F042A4220CD530ECA7B04FA7B04C3CF042A4220CD532A
:1E0474000ECA7B04FACF041182212A1220EB7E470E01B723CA9E047EBAC29704237EFB
:1E0492002BBBC29E04FF0E0C05C28B0479FE09D2A01A3282217 2237323E5211A20EF58
:1E04B000E1F7E5211620EFE1F7EB2A63212BEB722373233A6221 77233A612177C3F438
:1E04CE00012A61215E1600197EB7CAAC1A226121FF03CF11451FD7C2CF04CF3A13207F
:1E04EC00BEC2CF043A122023FE20CAFD04BEC2CF04CF7EB7C2CF04C3F401CD0714E5B6
:1E050A003AF5213215 20FEE7C2240521052011 58201A3C47CD241CC3280521 1620F795
:1E052800E1AF4F477E04FE3DC235050C23FE3EC23D050C0C23FE3CC2470579C6044F52
:1E0546002379B7C5CAA01AC132142078FE02C22C05CD0714224220 3AF521211520BE99
:1E056400C2A01AFEE7CAC505211620CD530E3A14201FD27F05CD750FCA9E053A1420F0
:1E058200E602CA8D05CD750FFA9E053A1420E604CAF401CD750FFAF401CAF4012A42A7
:1E05A00020116D1FD7CAE8022A4220114C1FD7CA47032A4220113 61ED7C2A01ACD35BC
```

:1E05DC003E20121ABEDA0406C209067830FAED0547793DFAF3054F78B1C2CE05060160
:1E05FA003A142ĿA0CAF401C39E050602C3FA050604C3FA05CD4008C5D5CFFE3DCA296E
:1E061800063A6921B7CAA01A21641DCD9119C3B60023CD0714CD681AE1D17BB73AF5C0
:1E0636002 1FA4306FEE3C2A01AF7C3F401FEE7C2A01ACD4E06C3F4011158201A96CAF9
:1E065400A306545D7E3C133DC25A0613131A4F131A47C52B7EB7CA8106E52B2B462BDD
:1E0672004EE17E122B1B0378B1C27406C36806EBC1702B71115820 1A06FFC605CA9641
:1E06900006D2960606FE2F3C4F2B702B71FF03093600231A3C471A77231305C2A606A8
:1E06AE00C9CDD41BCD681AEB221220E5218221467 8B7CAAC1A23D17E23BAC2D0067E08
:1E06CC00BBCAD906FF0D05C2C506C3AC1A3A821903C32822123E5CD3F1BEB22422069
:1E06EA00EFE1E5CD590D211620F7E1E5FF04CD530ECA1807E1E57EB73A4820FA14072A
:1E070800B7FA1807218221 35D1C3F401B7FA0C07E1FF0856235E237E326221237E32D5
:1E07260061 21EBCD681A2A612123227E21211620EF2A4220F7C3F40111691DE5D7CACB
:1E074400C207D121052022422 03600EBCFFE27CA5B07FE22C27E074F110520237EB9AC
:1E076200CA6A071213C35F0723EB36FECD8919EBCFFE2CCA7D07FE3BC27E0723CD405D
:1E0780008E5D52A42207EFE2C23CA92073E3FCDD518CF79B7FAB907CD1B0BCFFE2C03
:1E079E00CAA507B7C2B01A224220E1F7E17EFE2C23CA50072BCD681AC3F401CDFE0757
:1E07BC00224220C3AA07D1CDD41B7BB7F2A01ACD3F1BD5CD681A3E01326021 3E3FCD35
:1E07DA00D518060011592021052 07EB7CAF0070412 1323C3E40732602178325820E198
:1E07F800CD4E06C3F4011158200600 7EFE22CA1F08FE27CA1F08FE2CCA3208B7CA32F5
:1E0816000804131223CFC30E084F237EB9CA3008B7CAB01A041312C3200823CF115863
:1E083400207812D1EBE3D5CD4E06E1C9CDD41BD57AB7F25D08CD3F1B3EFF326821E3A2
:1E085200CD0714E3CDD20AEBE1C1C9CD3F1BC1C9CD4008E5D52AF6217EB7C28B082350
:1E0870007E22F621B7CAA81AFF0311381FD7CA8B082AF6215E160019C37008CF79B7CA
:1E088E00FAB308CD1B0B7EFE2CC2AC082322F621E1F7E17EFE2C23CA62082BC3B307CA
:1E08AC00B7CA9B08C3B01ACDFE077EFE2CCAC308B7C2AC08C3C4082322F621C3A008EB
:1E08CA00CD07147EFE2CC2A01A23CD3B1C110F20EB36D323772336C9EBCD0714CD68C5
:1E08E8001ACD3B1CCD0F20C3F4013A6621FE02C29800CDB41AC39101CD681A21D11E96
:1E090600CD9119CDFC1B3A6721B7C2980032652 12A61215E1600192 3227E211100209F
:1E092400CDDD19EB3600C39800CD681A3E01327C21116C21217421 0608CD241CC3F4F3
:1E0942 001CD0714CD3B1CB7CAA01A4F0DAF321420116D1FE5D7CA6909E1114C1FD715
:1E096000C2A01A3E0132142 0E5D179B7CA8B09CFB7CAA01AFE2CC27D0923C36A09CD72
:1E097E00351BC2870923C37D090DC26A09CD8C1AFE2CC2970923C38B09CD351BC2A013
:1E099C001ACDC01A7ECD811ACAAB0923C3A009CD681A3A1420B7C25003214522C3FBBA
:1E09BA0002CDD41B7BB7F2F909CD3F1BD5116F1FD7C2A01ACDD41B7AF6805 7CD3F1B23
:1E09D800FFF5D1EBCD681AEBD11A4704C5D5E5CD7012E1F7FFF8D1C1131A05C2E409D1
:1E09F600C3F4017AF6805 7CD3F1BFFF5E3116F1FD7C2A01ACDD41B7BB7F2A01ACD6825
:1E0A14001ACD3F1BE1D5115820D5EFFFF8E5CD3B1CE1D14704121 3C5D5EFFFF8E5CD2C
:1E0A32003B1CE1D1C105C2290AE1CD4E06C3F401CDD41BF2A01ACD3F1BE3F53EFF3252
:1E0A50006821CD0714F1E3D5C50313E5F521000019 0B78B1C2600AF1D1292901080002
:1E0A6E00FAAE0AE509EB2B2B7B2FC6015F7A2FCE00772B73EBE1444DEBD12B722B73D9
:1E0A8C00D12B722B732B36002B0B78B1C2920A3600E17EFE2CC2A80A23C3420ACD6888
:1E0AAA001AC3F401091B1B1A471B1A4F097CB7FAC40AC2A01A7DB7C2A01AEB2BD172AF
:1E0AC8002B73D12B722B73C39D0AD5FFFC562B5E7AB8DAA01AC2E50A7BB9DAA01A2BE3
:1E0AE600562B5EE3E5D513210000 78B1CAFA0A190BC3F00AD1C17AB8DAA01AC2090B3C
:1E0B04007BB9DAA01A092929 7D2FC6015F7C2FCE0057E119FFFCC9EB2148200605CD04
:1E0B2200331CCD8E0B0E00FE2EC2310BCD9E0B3A4C20814FFE45C2670BD5214C207192
:1E0B400023F72148200604CD331CD113CD8E0B3A4B204F3A4820B7F25D0BAF914F212D
:1E0B5E004C2079864F23D5EFD12148207EF61877CD1D0E79EBB7C8E521461EFA860BAB
:1E0B7C003DF5CD6C0EF1E1C3730B3CF5CDE30EC3810B1AFE2BCA9E0BFE2DC2A00B3E05
:1E0B9A0080324820131AFE30D8FE3AD00D3A4920E6F0CAB60B214C2034C39E0BCDCE0F
:1E0BB8000B1AE60F214B2086772B7ECE00772B7ECE0077C39E0BD5CDE90B215120F715
:1E0BD600CDE90BCDE90B114B202154200603CD950FD1C9214B20545D0603C3950F119C
:1E0BF4004B201AF60712CD750F3620F2040C362D23C20E0C363023362 0C93A4820CD2D
:1E0C1200830FC2190C3E80E680326A21E53A4820CD830FFE01F2400C216A2134214600
:1E0C30001EF23A0CCD6C0EC31F0CCDE30EC31F0CFE05F22A0C214C20F73A4820CD83CE
:1E0C4E000F0E06CDA40CFE0AFA600C214C20EFC32A0CCD920CAF324820CD880C214C83
:1E0C6C0020F7CD880CCD880C114B20214F200604CD950FCD920CC2630CC3BB0C214B78
:1E0C8A0020545D0604C3950FF630E1E3772379FE06C2A10C362E23E30DE95FAF3248B2
:1E0CA80020214B2006047E17772B05C2AE0C1DC2A90CC9E13645233A6A21362B57B7B0
:1E0CC600F2D20C362DE67F2F3C572F3C23E51EFF1CD60AF2D60CC60A477BCD920C78E1
:1E0CE400CD920CE136207AB7F2F50CFEFED8C3F80CFE06D04F060536202B05C2FB0C14
:1E0D0200EB7BD6056F7ADE006779B7CA1E0DFA330D46237E702B77230DC2130DEB7E91
:1E0D2000FE30C22B0D36202BC31F0DFE2E23C02B3620C9FEFFC2410D2B7E362E237710
:1E0D3E00C31E0D2B7E36302377626B06062B7E23772B05C24B0D362EC31E0D237EB7F0
:1E0D5C00CA750F2BCF750FCA28001148201ACD830F477FCD830F4F90CA890DF27C0DFD

```
:1E0D9800214D20F57891F2AE0D21482079E67F4F7EE680B17723F14F0603AFE5CDA0D5
:1E0DB6000FE10DC2B00D114820214C201AAEFAEF0D114B20214F200603CD950FD27512
:1E0DD4000F2148207EE680477ECD830F3CE67FB07723370603CDA00FC3750F215020E2
:1E0DF2000604CD241C114B20214F200603CD8A0FD21D0E114820214C200608CD2D1C2A
:1E0E100011482021 4C207EEE8012C3F70D214B207E2BB62BB6C22B0E2B77C97EB7FA98
:1E0E2E00750F23230603AF7E17772B05C2350E7ECD830F3DFE80CA981AE67F477EE6BF
:1E0E4C0080B07723C32B0E237EB7CA750F2B114C200605CD2D1C214C207EEE8077C3A0
:1E0E6A00590DCD750FC8237E2BB7CA2800114820 1ACD830F477ECD830F80CD650FE6B9
:1E0E88007F471AAEE680B0121155200603 23CD2D1C214C200606CD331C114920060396
:1E0EA6001A77AF12132305C2A60E0E180603215520AF7E1F772305C2B80ED2CE0E1134
:1E0EC4004E202154200606CD950F0606215420AF7E17772B05C2D40E0DC2B20EC31DF1
:1E0EE2000ECD750FC8237E2BB7CA981A7ECD830F47114820 1ACD830F903CCD650FE6C3
:1E0F00007F471AAEE680B012E513214C2036002306031A77AF12231305C2120FD106EF
:1E0F1E0003133600 23CD241C0E18114F2021532006 04CD8A0FD2420F114F2021532058
:1E0F3C000604CD950F373F0603214B207E17772B05C2480FAF0604214F207E17772B3C
:1E0F5A0005C2560F0DC2280FC31D0EB7F26F0FFEC1D0C3981AFE40D8C3981A3A492000
:1E0F7800B7C83A4820F67F3A4820C9E67FC640EE40C9AF1A9E121B2B05C28B0FC9AF60
:1E0F96001A8E121B2B05C2960FC97E1F772305C2A00FC9AFF5CD750FCAB70FF2BB0F51
:1E0FB400CD2A11F1EE80F521821ECD530ECAD00FFAD00F21821ECD530EC3B70F218237
:1E0FD2001ECD590DF1FCDA11211E20F73A4820CD830FF2EA0FFEFDD821821ECDE30E44
:1E0FF000212220F7212220CD6C0E212620F7212220CD6C0E212220F706042 1861EC5E9
:1E100E00E5CD6C0E211E20CD590DE1C105C8FF04C5E5211E20F7EFCD6C0E212220F704
:1E102C00E1E5C30F1021821ECD590DC3A90F213620F7CD3110214820113620 06044ECB
:1E104A001A7791223130 5C24910CDA90F213620CDE30EC9CD750FF26D10CDDA11CD4E
:1E1068006D10C3DA11213E1ECD590D211E20F721421ECD530E211E20CDE30E2122200A
:1E108600F7212220CD6C0E212620F711621E211E200604CD241CEF060721661EC30DD5
:1E10A40010CD750FFA981A212220F73A4820CD830FF2C4102FC602CD7012CDDA11C33F
:1E10C200C910DE01CD701221521ECD6C0E211E20F7EF3E01324820213E1ECD530E2147
:1E10E0002220F7F7060721961EC30D10CD750FF20611CDDA11CD061121 1E20F7213E55
:1E10FE001EEF211E20C3E30E212220F7F7213E1E111E200604CD2D1C060621B21ECDAD
:1E111C000D10214820CD6C0E214820C36C0E3A4820E67F324820C9CD750FC8E680F623
:1E113A0001F5213E1EEFF1324820C92148207EE640CA53110604C3331C7EB7F26211D0
:1E115800214E1ECD590D2148207EE63FFE18F0473E18904F214920AF06037E1F772392
:1E1176005C272110DC26C1121 48207EE680C61877C31D0ECD750FC8FA981A326A2193
:1E119400AF324820212220F721561ECD6C0E215A1ECD590D211E20F7CDC411211E209B
:1E11B200F7CDC4113A6A211F324820D0215E1EC36C0E212220EF211E20CDE30E211EB0
:1E11D00020CD590DD601324820C9CD750FC8EE80324820C906042 14820CD331C0E03CA
:1E11EE00214920E521792106063 77E17772B05C2F811231 1561D06031ABE1323DA1542
:1E120C0012C221121ABED22112EB1A96121B2B1A9E121323EB132305C2061206 0311D3
:1E122A007B211A86121B2B1A8EE63F122B05C229123E08911FE13A7B217723E50DC20B
:1E124800F211E13A7C21B7CA1D0E116C2121742106 08CD2D1CC31D0ECD3B1C210F2047
:1E126600 36DB23772336C9CD0F205F1600C3471D3A6B21C37012D11158201A4F060031
:1E128400EB09EB8646D292123EFF91473EFF3258 2078B7CAA21223137E1205C29A1249
:1E12A200E12B3A58201F3CEB2A59214F06000922 5921C1210000E53DC2B81239EB7266
:1E12C000237323 36E7E5215820 7E3C4F7E1213230DC2CC12E1FFF93E04CDED1AC35E30
:1E12DE00163A5820C37012CD3B1C2158203601 2377C92158207EB7CA7012237EC370A0
:1E12FC00122158203 6023CDF30BAF472B04BEC20813 70C92158207EB747CA7012119A
:1E131A00582023 7EFE20CA2513121305C21C13AF1221 5820CD1B0B7EB7C2B01AC9CDBD
:1E1338003B1C21582077B7C8233620 3DC33F13CD3B1C325820CD721C2158204623 7744
:1E1356005C25413C9CD721C4F0601C37713CD721C470EFFC37713CD721C47C5CD72E2
:1E137400 1CC14F7821582 0BEDA8513CA851336 00C981DA9213DE01DA9213BEDA9613EE
:1E1392007E903C4F715816001911 58207E1323120DC29E13C9CD721C215820B7C2B6F1
:1E13B00013360 0C3BD13BECABD13D2B1134F06007E913C0947E52A4220 2356235E23D7
:1E13CE0022422 0E11AB7C2DD130E0179C37012EB78BEDA0214C5D5E54EEB131ABEC2D6
:1E13EC00F913230DC2E813E1E1C1C3D913D1E1C10C2305C2DD130E00C3D913AF325869
:1E140A0021EB21000022592 12AFA21233600224020EBCFCD351BC23614CD1B0B444D74
:1E1428002A4020CD0B1B2240206269C3B215FE2ECA2314CD2C1BC226154623 0E20CDB2
:1E14460 0351BC27B14234FCFFE24F5C25A1479F6804F23CFFE28CA7E18E55059CD3F0F
:1E1464001B2A4020CD241B224020EBE1F1C2B215EB36E7EBC3B215CD2C1BC24D142B12
:1E1482007EFE20CA8114E5118E1DD7CAFC14E1E511BE1FD7CAD714E1E5113B1ED7CAEE
:1E14A0000A15E111721DE51AB7CACF14D7CABB14CD041B131313 13E1C3A614CD041B39
:1E14BE00131A4F131A47CFFE28C2A01A131AC31415E1460E2023C34D14D1CDD41B422B
:1E14DC004BEB2A4020237023712336AF79B7F2EF1436CF224020EBCFFE28C2A01AC338
:1E14FA001C14FE28CAA214E5CDE411E1D1C326143CD1F5E5114A1EC36514D1EB2A40E6
:1E15180020237023712377224020EBC31C14FF2DC23A15EB2A4020233661224020EB9C
```

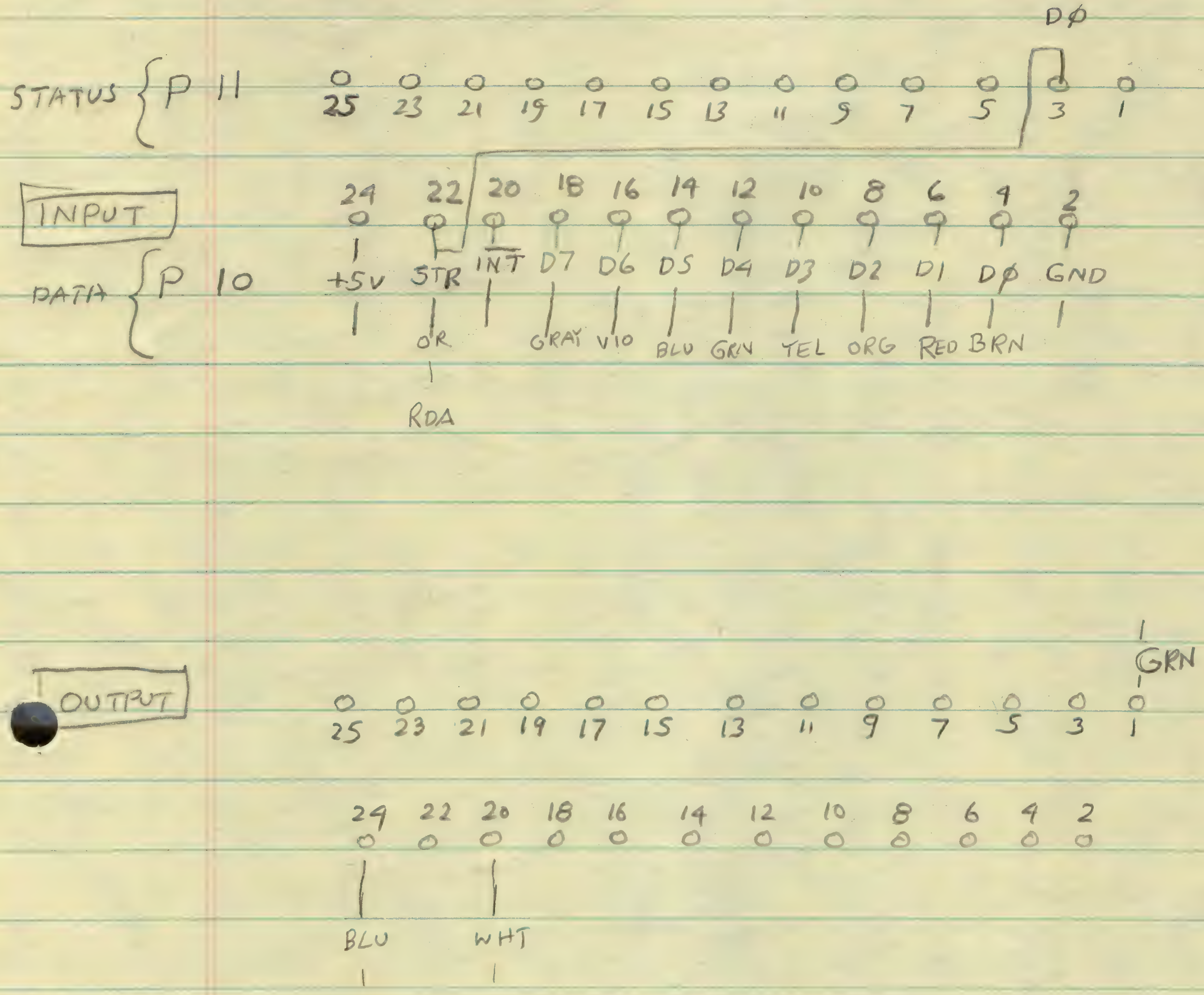
```
:1E155400237E1312B7CAA01A04B9C2541523115820781 21F3C4F0600E52A59210922F6
:1E15720059 21D121 0000E50DC2781539D5EB2A4020237223732336E724020EB1158EA
:1E159000201A3C471A77231305C29415E1C3B215EB215821342A40202336052240 20BB
:1E15AE00EBC33615CFFE2B0621CAED15FE2D0625CAED15FE2F0645CAED15FE5E0681F2
:1E15CC00CAED15FE29CA5016FE2CCA3B16FE2A0641CAED153A5821B7C2A01A224620F0
:1E15EA00C35E16E511010 02A40 2078E6E34F7EE6E3B9DA1816CA0D16E603132B3DC280
:1E1608000416C3F815E620CA17162B13C32A167EFE61CA2A16E6E0CA2A16110400 2AA6
:1E1626004020FFFD234E70411B7AB3C22A16224020E1C336153A5821B7CAE015EB2A2F
:1E1644004020233601224020EBC336153A58213D325821FAA01A232246202A402001CE
:1E166200000047E2BFEE3C274162B2B04040CC36416FEAFCA781CFECFCA781CF5E6D8
:1E168000E3FEA3CA9416FEC3CA9416F1FEE7CA6C16C31C1723224220D1C52B5E2B56C0
:1E169E0022442021AB16E5D5CD631CE1E92A44202 3237EE6E0FEC0CAF016C12A5921EB
:1E16BC002323225921210000E5E539C5E5F73EE3D12A4220772B732B722A4420232365
:1E16DA007E46FFFD78C1E6181F1F1F578282 0404CDED1AC35E16115820 1A1F3C2A59AF
:1E16F800214F0600092259212 10000C1E53DC2041739C5E51A3C471A77132305C20FBB
:1E171600173EE7C3CC16FE05C23A173E01CDED1A2A46203A6821B7CAB2153A5821B7A6
:1E173400C2B215C35E16B7C26B173A6821B7C4401879FE01C29C1A232356235ECD2646
:1E175200182A59217DB4CA6017D12BC356173A6821B7C467182A4620C9FE2111BF1703
:1E177000CA9D17FE2511530ECA9D17FE41116C0ECA9D17FE4511E30ECA9D17FE01CAF6
:1E178E001A18FE61CA0618FE8111C717C29C1A2323C5D556235EE5CD2618E1232356BA
:1E17AC00235E233AF521BEC2A01A2BE301F517C5E5EBC9FEE7CA7C12C3590DE5211EEE
:1E17CA0020F7E1EF213620F721421ECD530ECAEB17211E20EFCDA510213620CD6C0EA3
:1E17E800C3EC10211E20EF211E20C36C0EE1C12B2BCD0B1BFFF93E04CDED1AC35E160A
:1E18060023C5E52356235ECD2618CDDA11E1C1CD0B1BFFFC3E01CDED1A215B2134C303
:1E1824005E16237E32F521EBFEE7C228001158207E3C477E1223130 5C23718C979E502
:1E184200FE01C25318060421 1E20CD331CE10E01C9FE02C2A01AFF0556235ECD2618BC
:1E186000211E20F7C34F18CD3B1C4F0600C5211E20EFCD3B1C5F1600C1AF326821C9D1
:1E187E0078F68047C5EB3A5821F5AF3258212A5921E5210000 2259212A4020E5233657
:1E189C0000224020 3A6821F5EB3EFF326821CD1C14F13268212 24620E1224020E1227A
:1E18BA005921F1325821E1D5C5EBCD3F1BD1C1D1CDD20AEB2A4620E5C36514323E2035
:1E18D80021042036002 33A66211FDAF5183A3E20CD2719FE3FC2F5183E20CD2719DB96
:1E18F60003E602CAF518CD131A77FE0ACAF518FE0DC249193A66211FD4321936003A24
:1E1914006021B7C02B7EFE20CA1119B7CAD818210520C9F5DB031FD22819F1D302C9EE
:1E1932003E0DCD27193E0ACD27193EFFCD2719CD2719AF326B21C9FE15C25719CD410A
:1E1950001ACD3219C3D818FE7FC27A193A66211FDAF5182B7EB7CA85193E5CCD27198C
:1E196E007ECD27193E5CCD2719C3F5183A66211FDA85197ECD271923C3F5183AF32135
:1E198C00B7C021052 07EFEFEC8FEFDC2A019CD3219C3AF19B7CA32197ECD27193A6B29
:1E19AA00213C326B2123C391193AF321B7C03A6B21FE38D2321947AFB8CAC919D2CE9C
:1E19C80019C60EC3219326B2190473E20CD271905C8C3D319C50601CDE819CDE81992
:1E19E600C1C97EE6F01F1F1F1FC2F619B0C2FC190600F63012137EE60FC2061AB0C219
:1E1A0400C1A0600F630121323C9DB03E602C8DB02E67FFE03C2321ACD411A3A692196
:1E1A2200B7C29101 21D11ECD9119CDFC1BC39101FE0FC0CD411A3AF321EE0132F32163
:1E1A4000C9F53E5ECD2719F1C640C32719B7C2551AAF7EC1C9BECA621A78FE03D251E8
:1E1A5E001AC3521A041323C31300CFCD811AC2A01AFE3ACA791AFE5CC27C1A3265215F
:1E1A7C0023226321C9B7C8FE3AC8FE5CC8FE21C9CFCD811ACAA01AC9014C55DF0156D5
:1E1A9A004FDF015453DF014E53DF015452DF014144DF01584EDF015643DF06193EFFB2
:1E1AB800CD271905C8C3B61A0100003E0432F4217ECD351BC0E60F573AF4213DFAA04C
:1E1AD6001A32F4211E047917 4F7817471DC2DC1A79B24F23C3C81AE55F160019EBE164
:1E1AF4002F3C80471323 1A7705C2F81A224020C91AB7C813C3041BEBE3E5E521000070
:1E1B120039 2323C5D5E5F7D12A5921232 2225921E1C1237223732336E3C97EFE41D802
:1E1B3000FE5AC33B1B7EFE30D8FE39C8D0BFC9E53A7D21B7C29A1B2AF8217EB7CA70AE
:1E1B4E001B2B2B462B4EFF037EBAC26C1B2B7E23BBC26C1B7AB7FACF1B0923EBE1C920
:1E1B6C0009C34A1B722B732B7AB7FAAE1B36FF2B7BB7FA881B36F80604C38C1B36FBF9
:1E1B8A0006012B360005C28C1B2B360023EBE1C9215C217EBAC2471B237EBBC2471BD4
:1E1BA8002356235EE1C9E536FE2B36142B36002B360A2B36002B360A01E5012B36000D
:1E1BC6000B78B1C2C31BE12323C1D1C5D5C9CFCD2C1BC2A01A571E2023CFCD351BC24C
:1E1BE400E91B5F23CFFE24C2F41B7BF6805F23C9FE28C07AF68057C911002 02A7E216F
:1E1C0200CDDD19EB3600210020C39119712336FE210520CD911921E01ECD9119CDFC4E
:1E1C20001BC398001A77231305C2241CC9EBCD241CEBC936002305C2331CC9E5CD45BE
:1E1C3E0011FA601C3A4820CD830FFE09D2601C473E0890473A4920CA5E1CB71F05C2C3
:1E1C5C00581CE1C9AFE1C92A4220 2356235E232242202BC32618CD631CC33B1CC5D59A
:1E1C7A00E5EB2A4620E5EB2246202A5921E53A5821473A7D214FC53A6821F5AF326824
:1E1C9800212A5C21E52A5E21E52AFA21E52A4020E522FA213E01327D212A46204E2B05
:1E1CB600462145227EB7CAA01AE5FF0311BB1FD7C2D81CC5CDD41BC17AB8C2D81C7B85
:1E1CD400B9CAE01CE15E160019C3BA1CD1CFFE28C2A01A23CDD41BE5215C21722373C0
```

```
:1E1C7A00E5EB2A4620E5EB22462[illegible]
:1E1C9800212A5C21E52A5E21E52AFA21E52A4020E522FA213E01327D212A46204E2B05
:1E1CB6004621452227EB7CAA01AE5FF0311BB1FD7C2D81CC5CDD41BC17AB8C2D81C7B85
:1E1CD400B9CAE01CE15E160019C3BA1CD1CFFE28C2A01A23CDD41BE5215C21722373C0
:1E1CF20023EB2A462023237E1213237E12E1CFFE29C2A01A23CFFE3DC2A01A23CD07A7
:1E1D100014CD681AE1224020E122FA21E1225E21E1225C21F1326821C179327D2178A1
:1E1D2E00325821E1225921E1224620E1D12B224420FF05224220C3AB1621482036180
:1E1D4C00233600237223373C31D0E3FFD3FEB3FDD1BEC33D31A852B1E5748415400 4C10
:1E1D6A00494E4500544142004142530 02A11AB535152008A11AB494E54004511AB5371
:1E1D8800474E003311AB524E4400E411AB53494E00A90FAB434F5300311 0AB54414E34
:1E1DA600003A10AB41544E005E10AB494E50006012AB4C4F4700A510AB45585000EC0F
:1E1DC40010AB504F53007612AB4C454E00DF12AB4348522400E512CB415343494900 7A
:1E1DE200F012AB4E554D2400FD12CB56414C001013AB535041434524000713CB53544B
:1E1E000052494E4724004713D34C4546542400 5B13D35249474854240064 13D34D4931
:1E1E1E00424006D13DB494E53545200A713BB0013140053544550005448454E0050FC
:1E1E3C004900018000000280000004A0000002C90FD780FFFFFF00B17216009714EB9B
:1E1E5A007FD5A95601B504F300C90FD700FFFFB8FFAA95BC7ECAD520FE8782D67DA3D1
:1E1E7800131CFC89A6B879DF3A9E01C90FD780A55DDE7DA33455F9993860749ED7B688
:1E1E960000FFFEBCFFFFBAAC7FA80E28FEE74B507E89DEE0FCE1C5707AB03FA87EFFCF
:1E1EB400FFA47C8007F078A9D3F974B3E54E6FB6350B6CB961A8FD52454144590 0FD31
:1E1ED20053544F50204154204C494E4520FE204552524F5220494E204C494E4520FECA
:1E1EF0004C4953540051 02 52554E00CB014E4557009E01434F4E00DD02544150 0084D3
:1E1F0E00015341564500740 14B4559009101465245 00AD0149460006055245414400 54
:1E1F2C006208524553544F5245002303444154 4100F401464F52000 2044E45585400A8
:1E1F4A00AF06474F5355420047035245545552 4E002F03494E505554003E0750524928
:1E1F68004E540065 03474F544F00E8024C45540 00E0653544F500000009454E4400F21D
:1E1F860008 52454D00F4012100F4013F006503524 14E444F4D495A45002D094F4E0023
:1E1FA40043094F555400CA0844494D00420A4348414E474500BB0944454 6464E00F422
:1E1FC2000100000000000000000000000200000000000000000000000000000000000000
:1E1FE000000000000000000000000000000000000000000000000000000000000000E3
:021FFE000000E1
:00000001FF
```

```
44        272
66         31
66        ----
66        272
30        816
----      ----
272       8432
```

Left = 0 (L 1/2)
R = I (L 1/2)

Steve Kloch
834-0912



OP—80A

I/O SOCKET

PIN NØ. 4 (I10)		DØ	1	BRN	RED	16	D1	PIN NØ 6 (I 11)
	8	D2	2	ORG	YEL	15	D3	10
	12	D4	3	GRN	BLU	14	D5	14
	16	D6	4	VIO	GRY	13	D7	18
$\overline{INT}$	20	ACK or $\overline{ACK}$	5	WHT	BLK	12	SPARE	
		$\overline{RDA}$	6	BRN	RED	11	S2	
STB	22	RDA	7	ORG	YEL	10	S1	
		GROUND	8	GRN	BLU	9	+5vdc	

DØ thru D7 = DATA OUTPUT BYTE

S1 and S2 = STATUS LEDS

RDA = READER DATA AVAILABLE (_⊓_)

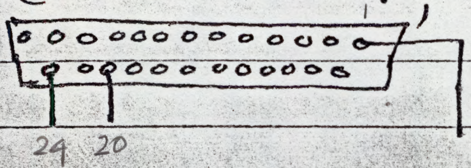
$\overline{RDA}$ = READER DATA AVAILABLE (‾⊔‾)

ACK or $\overline{ACK}$ = ACKNOWLEDGE (Resets RDA and $\overline{RDA}$) (_⊓_) or (‾⊔‾)*

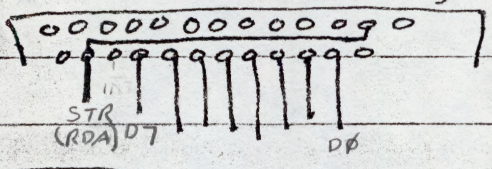
POWER = +5vdc @ 175ma MAXIMUM

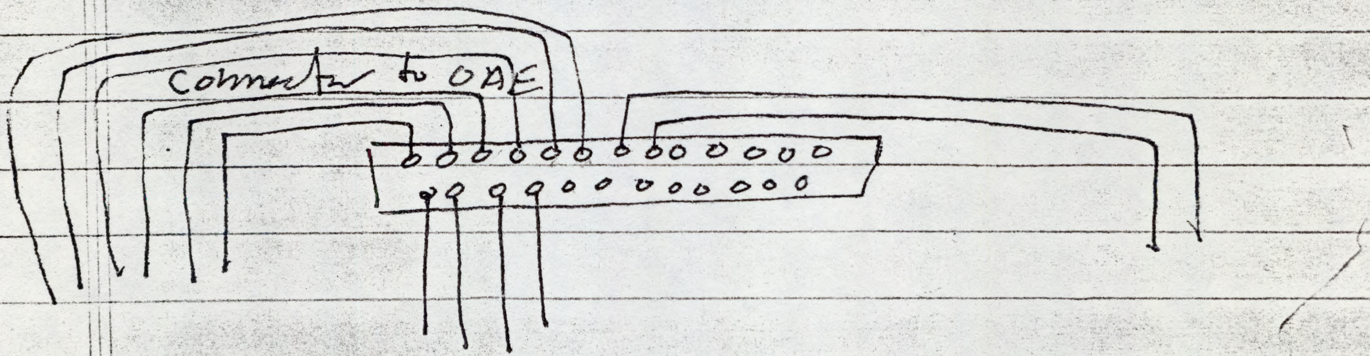
OAE Tape reader

PIO ~~OUT~~ OUTput
(Data from Computer)



PIO INPUT
(DATA TO Computer)





Steve Klock
1116 N. Westwood
Mesa, Arizona 85201

DØ	→	I25
D2		
D4		
D6		
$\overline{\text{ACK}}$		
RDA		

DØ	BRN	→	I 25
D1	RED	→	I 24
D2	ORG	→	I 23
D3	YEL	→	I 22
D4	GRN	→	I 21
D5	BLU	→	I 20
D6	VIO	→	I 19
D7	GRY	→	I 18
$\overline{\text{ACK}}$	WHT	→	O 15
$\overline{\text{RDA}}$	BRN	→	I 15
GND	GRN	→	O 13
~~[illegible]~~ +5	BLU	→	O 14